# Surface Reconstruction from Distributed Angle Measurements

Thesis by
Thibaud Talon

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

Caltech

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2020
Defended December 6th, 2019

© 2020

Thibaud Talon
ORCID: 0000-0002-8240-1101

All rights reserved

# ACKNOWLEDGEMENTS

Many people have contributed, directly or indirectly, to the work presented in this thesis and it would not have been possible without their help, support and motivation.

First and foremost, I would like to thank my advisor, Prof. Sergio Pellegrino. He has been a great mentor, guiding me throughout my research, and always pushing me forward. I would also like to thank my committee: Prof. Ali Hajimiri, Prof. Dan Meiron, and Prof. Soon-Jo Chung for their helpful advice and comments during my candidacy and thesis defense.

I would like to thank Northrop Grumman and the Space Solar Power Project at Caltech for their financial support.

I had the chance to work alongside amazing colleagues in the Space Structures Laboratory who gave me great feedback during our group meetings and daily conversations. To that end, many thanks to Alan Truong, Alexander Wen, Andrew Lee, Antonio Pedivellano, Armanj Hasanyan, Ashish Goel, Charles Dorn, Charles Sommer, Christophe Leclerc, Corentin Lubeigt, Daniel Turk, Fabien Royer, Federico Bosi, Harsha Reddy, Ignacio Maqueda, John Steeves, Kathryn Jackson, Kristina Hogstrom, Lee Wilson, Luke Chen, Manan Arya, Maria Sakovsky, Mélanie Delapierre, Michael Marshall, Miguel Bessa, Nicolas Lee, Serena Ferraro, Steve Bongiorno, Terry Gdoutos, Uba Ubamanyu, Wilfried Jahn, Yang Li, and Yuchen Wei. I would also like to thank every member of the Caltech Space Solar Power Project team from which I learned a lot about cutting edge research in electronics, antennas, and photovoltaics.

I feel fortunate to have learned about aerospace from engineers working at NASA's Jet Propulsion Laboratory. I would like to thank Greg Davis, Anthony Freeman, Dan Scharf, Oscar Alvarez-Salazar, Andy Klesh, Scott Ploen, John Baker, and Jim Breckinridge. This includes everything I learned by working on the AAReST project (Autonomous Assembly of a Reconfigurable Space Telescope) during the entirety of my PhD. I would like to thank Prof. Craig Underwood and Prof. Chris Bridges from the University of Surrey for everything they taught me about building small satellites and mostly on how to design space software and avionics.

While AAReST was the main side project during my PhD, I had the chance to be the co-chair of the 2017 Caltech Space Challenge. I would like to thank Ilana Gat with whom I have organized the event, our advisors Prof. Paul Dimotakis, Jakob van Zyl, and Tony Freeman, our student experts Erika Figueroa Schibber, Francesca Baldini, Jane Zhang, Maria Sakovsky, and Yuchen Wei, every past chair, everyone at Caltech and JPL who helped organizing this challenge, our sponsors, and of course, all the participants who made the event incredible.

I would like to thank Michelle Judd, Tom Prince and everyone at the Keck Institute for Space Studies for welcoming me into their family and giving me the opportunity to meet amazing, successful women and men, listen to their story, and learn from their experience.

Everyone at the Graduate Aerospace Laboratories (GALCIT) has been a great part of my experience at Caltech. I specifically want to thank Petros Arakelian, Kate Jackson, Dimity Nelson, Peggy Blue, Jamie Sei, Christine Ramirez, and Jackie Gish.

I want to thank my friends from all over the world: from my high school friends, my friends from Polytechnique and all my friends here in Los Angeles who made this city feel like home.

My family has always been of great support throughout my studies and they have always been encouraging especially during this PhD. I want to thank my parents, Lionel and Isabelle, my brother Adrien, my cousins, aunts, uncles and my grandparents Paulette, Jacques, Jeannine and Pierre.

Last but not least, I want to thank Lauren for her understanding, patience, and encouragement and for sharing this journey with me.

# ABSTRACT

This thesis presents an innovative solution to the shape measurement of large structures for space applications. The current state-of-the-art heavily relies on optical solutions such as cameras or lasers to recover the shape of a surface. Because of the impracticality of placing a system in front of a large structure flying in space, new solutions need to be developed. The proposed solution is to embed angular sensors (such as sun sensors) directly on the surface. The sensors provide a collection of distributed measurements that form a discrete map of the angular orientation of the structure. An integration scheme can then estimate the 3D shape of the surface.

A mathematical model to perform the integration from angle measurements to the shape of a 3D surface is presented first. This model is purely geometric and serves as a basis for similar concepts. The surface is known in a reference configuration and is assumed to have deformed inextensibly to its current shape. Inextensibility conditions are enforced through a discretization of the metric tensor generating a finite number of constraints. This model parameterizes the shape of the surface using a small number of unknowns, and thus requires a small number of sensors. We study the singularities of the equations and derive necessary conditions for the problem to be well-posed. The limitations of the algorithm are highlighted. Simulations are performed on developable surfaces to analyze the performance of the method and to show the influence of the parameters used in the algorithm. Optimal schemes which lower the RMS error between the reconstructed shape and the actual one are presented.

An experimental validation of the proposed solution and algorithm is performed on a $1.3 \times 0.25$ m structure with 14 embedded sun sensors. The sensors measure the two local angles of the surface from a light source placed in front of the surface. A small, lightweight and expandable design of the sensors is shown in this thesis. A calibration procedure accurately correlates the output of the sensor with a $0.5°$ precision. The procedure also highlights the limitations of the design. The structure was deformed in bending and torsion with amplitudes of a few centimeters, and its shape was reconstructed to an accuracy on the order of a millimeter.

The accuracy of the initial algorithm is found to be limited by local shape deformations caused by the mechanical response of the structure. A new algorithm, replacing the discrete inextensibility conditions with the equilibrium equations derived from a finite-element model, is shown. This new algorithm is tested on the experimental structure and the accuracy of the reconstruction is increased by a factor of 2. The RMS error is under a millimeter on average over the different applied shapes and goes as low as 0.3 mm.

To understand how this solution can apply to large space structures, simulations are performed on a model of a large planar spacecraft. A 25×25 m structure representing the current concept for the Caltech Space Solar Power Project satellite is used as an example. Sensors with similar noise properties as the ones built for the experiment are placed on the spacecraft. A finite-element model combining the vibration of the spacecraft with large rigid body rotations is presented. This model is used in a Kalman filter that estimates the shape of the structure by iterative prediction from the dynamic finite-element model and correction from the angle measurements. Simulations are performed around the thruster actuation applied at the corner of the structure to follow a specific guidance scheme that is optimal for space solar power satellites. The actuation creates both vibrations of the structure with amplitudes of few centimeters and large rotations of the spacecraft. The designed Kalman filter can accurately estimate both effects and it is shown that millimeter accuracy is achievable. The relationship between the number of sensors, the reconstructed shape error, as well as potential stiffness deviations in the FE model is studied. The results provide first order estimates of the performance of this measurement system, in order to enable the design of future space missions.

# PUBLISHED CONTENT AND CONTRIBUTIONS

[1] **Talon, T.** and S. Pellegrino. “Inextensible Surface Reconstruction from Distributed Angle Measurements.” Manuscript submitted for publication. 2019.

*T. Talon developed the shape reconstruction algorithm, performed the singularity analyses, studied the performance of the algorithm through simulations, and wrote the manuscript.*

[2] **Talon, T.** and S. Pellegrino. “Shape Measurement of Large Structures in Space: Experiments.” In: *2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. IEEE. 2018, pp. 581–584. DOI: 10.1109/MetroAeroSpace.2018.8453578.

*T. Talon developed the shape reconstruction methodology, designed and fabricated the sun sensors, performed their calibration, conducted the experiments, and wrote the manuscript.*

[3] **Talon, T.** and S. Pellegrino. “In-Space Shape Measurement of Large Planar Structures.” In: *4th AIAA Spacecraft Structures Conference*. 2017, p. 1116. DOI: 10.2514/6.2017-1116.

*T. Talon developed the shape reconstruction methodology, the finite-element approach and study, the sensors location optimization method, built the sun sensors, designed and conducted the experiments, and wrote the manuscript.*

# TABLE OF CONTENTS

# LIST OF ILLUSTRATIONS

*Chapter 1*

# INTRODUCTION

## 1.1 Motivation

In his book *Surfaces and Their Measurement* [93], David J. Whitehouse categorizes the usefulness of surface measurement from two roles:

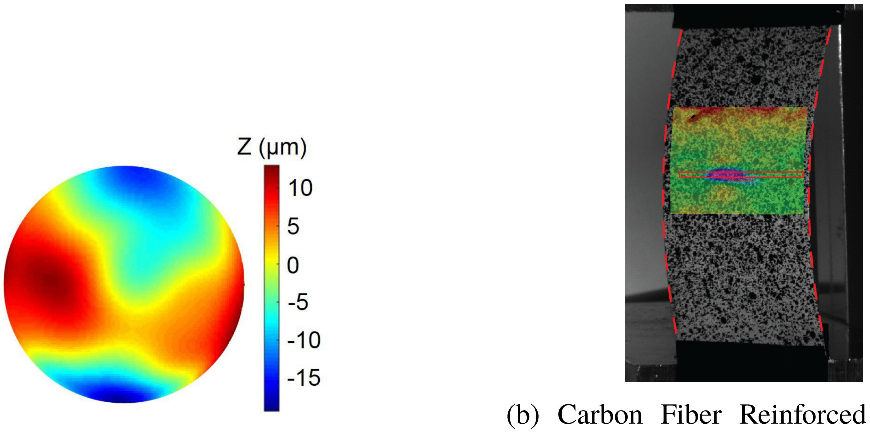
> *one to help control the manufacture, [...] and the other to help optimize the function.*

While this book mainly focuses on manufacturing and surface quality which is usually referred as the field of surface metrology, this quote is valid for many more applications.

An example of shape measurement *helping control the manufacture* are microsopic shape measurements to measure surface quality in applications such as optics [16]. Material testing also makes great use of shape measurement with samples ranging from few millimeters to meters. These tests characterize the mechanical properties of materials or the response of structures, and eventually assist in the design of more complex geometries [48, 64]. Examples of such measurements can be seen in figure 1.1. Measuring the shape of a mirror (figure 1.1a) can help identify manufacturing errors in order to correct the process. Figure 1.1b shows the material characterization of a carbon fiber sample. The results of the test will help design complex structures.

A wide range of applications falls into the second category, surface reconstruction to *help optimize the function*. Shape measurements are used in entertainment, gaming, security, etc [78, 96, 98, 99]. Figure 1.2 shows some examples of such applications. Body tracking using gaming hardware shown in figure 1.2a can recreate human motion and expression which can be integrated in immersive virtual environments. Face recognition techniques such as the one shown in figure 1.2b has applications in biometric security.

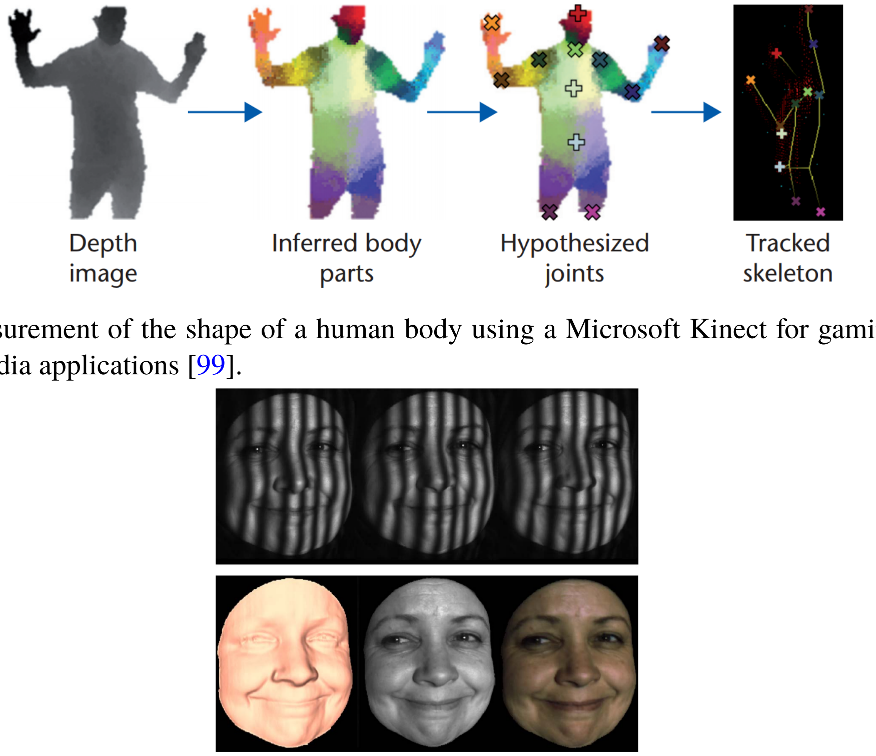
Historically, optical methods have been the major tool used to perform shape measurements. An extensive list of techniques can be found in [18]. Laser scanning, interferometry, photogrammetry, digital image correlation [83], structured light are among the many methods that have been studied. Field of view, accuracy and prac-



(a) Shape of a deformable mirror measured using a reverse Hartmann procedure [82].

(b) Carbon Fiber Reinforced Polymer (CFRP) sample under mechanical bending test. Its shape is measured using a digital image correlation system [92].

Figure 1.1: Examples of shape measurements helping the manufacturing process.

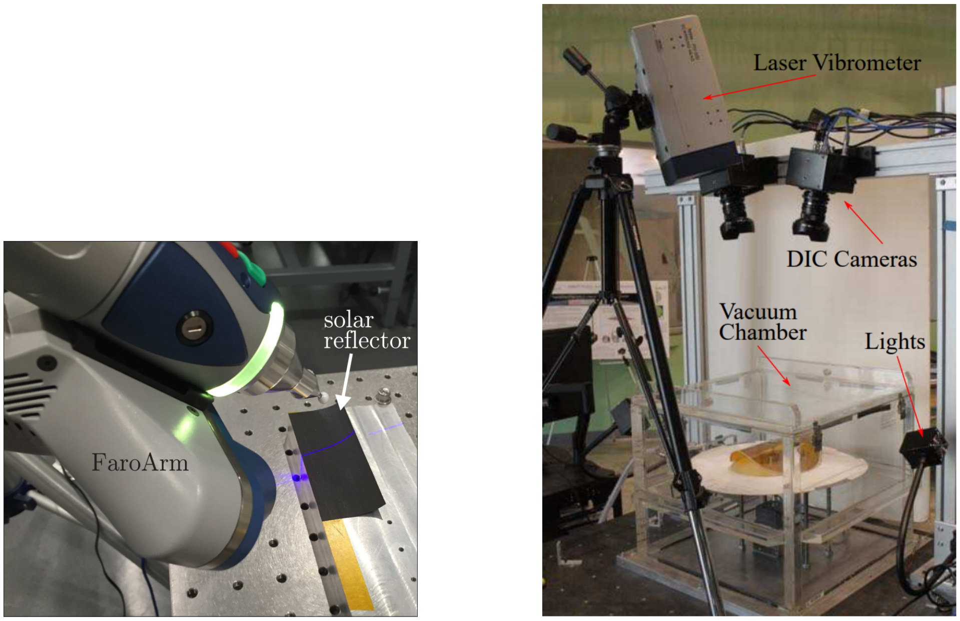


(a) Measurement of the shape of a human body using a Microsoft Kinect for gaming and multimedia applications [99].

(b) Real-time shape detection of a face for potential security applications [98].

Figure 1.2: Examples of shape measurements providing the functionality of gaming and security applications.

ticality are one of the main drivers to choose the appropriate technology. Micron level accuracy is now standard for many systems. Figure 1.3 shows two examples of measurement techniques. The scanning laser (figure 1.3a) measures the position of points on a structure. The output point cloud can then be integrated into a surface. Digital image correlation systems (figure 1.3b) use photos of a structure from different angles to estimate the position of features (usually painted speckles) from triangulation.



(a) Measurement of the shape of a solar reflector using a scanning laser [12].

(b) Digital image correlation (DIC) setup to measure the shape of rotating membrane [22]. A laser vibrometer is also used to measure the displacement of the edge of the structure.

Figure 1.3: Optical shape measurement systems commonly used for relatively large structures.

One of the limitations of optical systems is the work space required to perform the shape measurement. Such systems require a certain depth of view to project light and record its scattered reflection. Their accuracy is also usually linked to the size of the surface: the larger the structures the larger the measurement error. As a result, their application to large structures involves a trade-off between the precision of the measurement and the space required in front of the structure.

Recent applications have highlighted these limitations. Large deployable structures for space application, for instance, have been of increasing interest over the last

(a) Conceptual image of IKAROS. Credit: JAXA.

(b) In-flight picture of IKAROS from an on-board camera located in the central hub. Credit: JAXA.

Figure 1.4: Images of the IKAROS mission.

decades. Large solar panels, antenna arrays, solar sails are a few examples of concepts being currently tested or designed [3, 30, 88]. The measurement of the shape of these large space structures can be critical for their application. The use of an optical system such as the ones described in the previous paragraph is extremely complicated due to the large dimensions of the structure and the fact that it is free flying in space. Figure 1.4b for example, shows an image of the solar sail IKAROS (figure 1.4a) taken from an on-board camera. The depth of view of the structure is too small to be utilized for shape reconstruction purposes.

Embedded systems provide an alternative solution for the estimation of the shape of a structure. Sensors are directly integrated in the surface instead of requiring an external system. This approach has been studied for different applications [10, 38–40, 80, 81]. The sensors usually measure the angles of the surface at discrete location from a fixed direction (such as the gravity vector of the Earth's magnetic field). For space applications, sun sensors provide angle measurements from the direction of the Sun.

## 1.2 Research Goals

The overall objective of this thesis is to show the feasibility, accuracy, and limitations of a system capable of measuring the shape of large structures for space applications using embedded, distributed angle measurements. Other applications may use the proposed solutions, but space structures are used as a driving example throughout the work detailed in the thesis.

Such a system requires advancement in the state-of-the-art in two main categories:

1. **Develop efficient algorithms that reconstruct the shape of a surface from distributed angle measurements.**

   Different solutions are investigated which present various pros and cons. A static, geometry-based algorithm is first presented. Its general formulation makes it suitable for many applications. A more powerful algorithm is presented next. This second algorithm is augmented with mechanical equilibrium equations that predict the response of the structure under different loads. This shows more accurate for structures with non-negligible bending stiffness. Finally a dynamic algorithm is implemented, accounting for inertial loads to better reconstruct the shape of a vibrating structure.

2. **Engineer sensors suited for space structures and design an experimental system.**

   State-of-the-art sensors are not suited for application in ultralight space structures and new solutions need to be engineered. Developing the design of an experimental system provides a deeper understanding of the challenges that need to be tackled to apply the presented solution to a future space mission.

## 1.3 Layout of the Thesis

The layout of the thesis shows the steps undertaken to address the research goals.

Chapter 2 of this thesis details a geometric algorithm to reconstruct the shape of a surface with embedded angle sensors. Because the measurements cannot determine the in-plane stretching of the surface, it is assumed that the deformation is inextensible. Such an assumption is valid for relatively thin structures with small bending stiffness. Inextensibility equations are derived from differential geometry and discretized over the structure. The measurement equations of the sensors are explicitly introduced and are applicable to most types of angular sensors, especially light sensors which are used in most of the thesis. Combining the inextensibility and measurement equations leads to a non-linear system of equations. By parameterizing the shape of the surface over a set of 2D basis functions, one can solve for their amplitude by solving these equations. Stability studies are conducted to understand the possible singularities associated with the algorithm. Finally, simulations are performed to show its performance when reconstructing cylindrical and conical shapes.

Chapter 3 shows an experimental validation of the algorithm presented in chapter 2. Light sensors are designed to perform the angle measurement. They are based on a quad-photodiode with a square aperture placed on top, effectively acting as a 4-pixel pinhole camera. This design creates an accurate sensor that is small, light and inexpensive. Details are provided about the design as well as the full calibration process of the sensors. The experimental structure consists of a 1.3×0.25 m aluminum sheet. A total of 14 sensors are embedded on the surface whose deformation is also measured with a camera system to evaluate the performance of the proposed solution. The accuracy of the system is on the order of a millimeter RMS. Such an accuracy is suitable if an antenna array functioning at a 10 GHz or less would be placed on the structure which is the case of the Space Solar Power Project. Noticeable increases in the error are present for complex shapes where the relative smoothness of the reconstruction introduced by the algorithm fails to capture local effects.

In order to capture the local deformation of the structure which limits the performance of the previous approach, a new algorithm is introduced in chapter 4. A structural-mechanics model of the structure is used instead of the inextensibility conditions. This model aims to accurately capture the shape of the surface subject to external loads. A finite-element model used to estimate the mechanical response of the structure is presented. The algorithm introduced in chapter 2 is generalized to generate a new, finite-element based shape reconstruction algorithm. Its performance is demonstrated on the experimental data from chapter 3 and the results are compared to the previous algorithm. A significant decrease in the RMS error is obtained which is primarily due to the ability to capture local deformations.

Chapter 5 expands on the previous finite-element based algorithm to reconstruct the dynamic shape of a structure. A Kalman filter is used to minimize the RMS error of the estimated dynamic shape of a surface. A simple modelization of the structure combines the local vibration of a plate-like satellite structure and the large rigid rotations needed for its operation in orbit. Sun sensors are placed on the spacecraft and integrated with the dynamic model of the structure using the Kalman filter formulation. Multiple studies are presented to analyze the accuracy of the system for different numbers of sensors or under different modelization errors that can be present in the finite-element model. The formulation is detailed for a conceptual 25×25 m solar power spacecraft developed as part of the Caltech Space Solar Power Project. It is shown that millimeter accuracy is achievable for such a large spacecraft which is within the requirements of the mission to use the 10 GHz antenna array.

*Chapter 2*

# INEXTENSIBLE SURFACE RECONSTRUCTION FROM DISTRIBUTED ANGLE MEASUREMENTS

Talon, T. and Pellegrino, S. (2019) “Inextensible Surface Reconstruction From Distributed Angle Measurements.” Manuscript submitted for publication.

## 2.1 Introduction

Recovering the shape of a 3D surface often requires a measurement system with a certain depth of view. Cameras, scanning lasers or radars are often used to generate a point cloud of the surface. Methods exist to convert this cloud into a more or less smooth 3D surface [8, 46].

Many solutions focus on reconstructing the shape of a surface from a set of measurements and a reference configuration (also called a template). The 3D shape of the reference is known either from a previous measurement or by construction of the structure. For instance, the surface depicted in Figure 2.1 is known to be flat as it is a sheet of paper and its dimensions are dictated by a predefined printed pattern. Monocular reconstruction of a surface is a well-known approach [15, 59, 73].

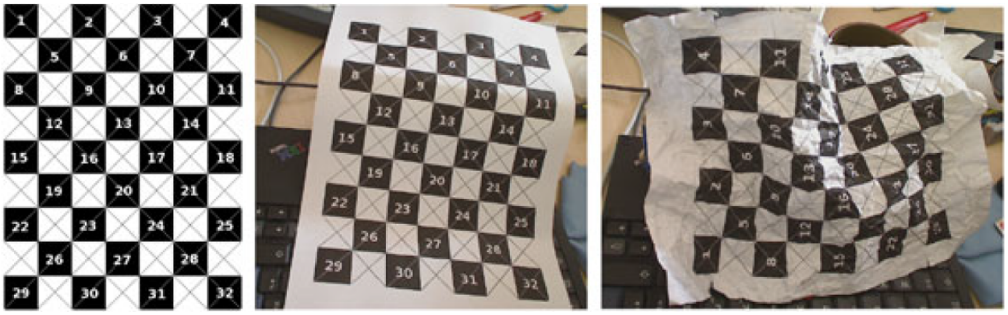


Figure 2.1: Example of images used on monocular, template-based reconstruction of inextensible surface [59]. The left image represents the reference configuration (a flat sheet of paper) and the other images represent two configurations of the sheet that will be reconstructed.

In some cases, these methods are not suited to measure the shape of a specific surface. For instance, the surface can be too large to accommodate the field of view or range of the system, there may be little or no space available in front of the structure, or the accuracy of the system may not be sufficient.

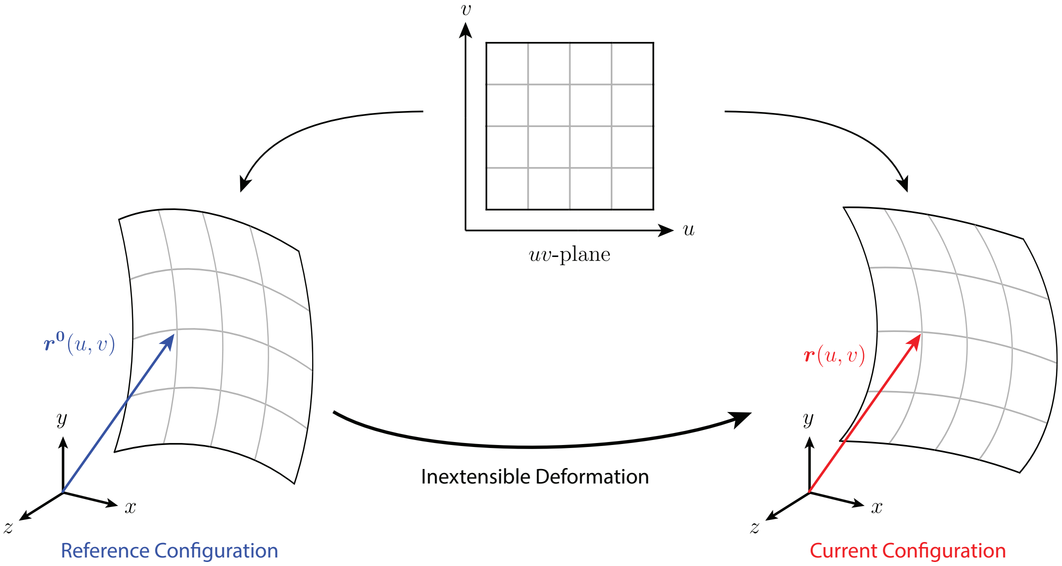


Figure 2.2: Definition of the problem. The surface is parametrized by two coordinates $(u, v)$. The 3D surface is a mapping of the 2D coordinates to 3D. The shape of the reference configuration is known while the current configuration needs to be reconstructed.

The approach investigated in this chapter is based on embedding angle sensors directly on the surface to be reconstructed.

Different technologies exist to perform such measurements. Most state-of-the-art technologies use inertial sensors (a combination of accelerometers and magnetometers) to measure angles from the gravity vector and Earth's magnetic North and have been investigated to reconstruct both 3D curves and surfaces [38–40, 80, 81]. Note that if the surface experiences large accelerations, the accelerometers fail to detect the direction of gravity. Magnetic fields are easily affected by magnets, currents, or ferrous materials, which limits the usage of magnetometers.

Sun sensors have recently been studied mostly for space applications where gravity is near-zero [10, 84, 85]. Different sensor technologies exist. The simplest ones use quad-photodiodes behind an aperture, effectively acting as a 4-pixel pinhole camera [85]. More complex architectures involve cameras with a large photosensor array that locate the centroid of the spot created by the light source using image-processing algorithms [87]. They can also identify features in the image (such as stars) to improve their accuracy.

The angle measurements of these sensors are fed in an algorithm that will be described in this chapter, and the algorithm reconstructs the shape of an inextensible support surface that holds the sensors.

This study is limited to inextensible deformations. This requires a reference configuration, also called template in the literature, to be fully known in order to define the conservation of lengths upon deformation. The overall problem is described in figure 2.2. Imposing inextensibility of the deformation of the surface usually serves two purposes: 1) to eliminate singularities in the algorithms and 2) to improve the results by adding some knowledge of the deformation. Many methods used to reconstruct inextensible surfaces employ a triangular [72, 73] or quadrilateral [33] mesh to map the surface. Each edge of the mesh can be defined as a straight, rigid line which implicitly enforces inextensibility. This method requires a fine mesh in order to achieve a smooth mapping; this means that many degrees of freedom must be computed from a large amount of data (for instance, high resolution images) which can be computationally expensive. Tangential and normal vectors are undefined at the intersection of edges which can cause issues in defining angles. Previous research involving embedded sensors does not strongly impose inextensibility of the deformation. The surface is reconstructed by integrating along inextensible lines where the angle sensors are placed uniformly [38, 80]. The inextensibility is either imposed explicitly, by enforcing conservation of lengths between sensors, or implicitly by connecting sensors by means of rigid lines. The surface is then filled by different techniques such as Coon's methods in [40] or using a quad mesh in [38]. While inextensibility is imposed along the lines of integration, the reconstructed surface usually stretches in between as the filling techniques only create a smooth surface without imposing inextensibility. The presented method improves the existing state-of-the-art by applying inextensibility conditions across the surface with only a limited number of equations.

Our approach is to reconstruct the shape of the surface in its current configuration by only assuming an inextensible transformation from the reference configuration (or template) and the measurement of angles at discrete locations along the structure.

In order to estimate the shape of a surface, we parameterize it on a set of basis functions. This is presented in section 2.2. Different sets of basis functions relevant to this problem are introduced. We define the inextensibility of the deformation in section 2.3 using the conservation of the metric tensor. The general mathematical relations are discretized in order to generate a finite set of constraints. The singularities associated with this first system of equations are analyzed. A set of angle measurements is defined in section 2.4. Their singularities are analyzed separately from the inextensibility conditions, and we especially investigate requirements on the placement of the sensors. The complete system of equations including inextensibility and angle measurement conditions is obtained in section 2.5. It is generally an overconstrained system that can be solved using the Levenberg-Marquardt algorithm. Singularities of this complete system are studied based on the remarks made on the singularities of each set of equations. Finally, the approach is tested in simulation focusing on developable surfaces, where the reference configuration can be a flat plane. We show the results of the reconstruction of a cylinder and a cone and investigate the influence of different parameters of the algorithm.

Table 2.1: Nomenclature

| | |
|---|---|
| $u, v$ | Curvilinear coordinates of a surface |
| $\boldsymbol{r}(u, v)$ | Position of the surface in the current configuration |
| $\boldsymbol{r}^{\mathbf{0}}(u, v)$ | Position of the surface in the reference configuration |
| $\frac{\partial \boldsymbol{r}}{\partial u}(u, v)$ | First tangential vector of the surface parameterized by $\boldsymbol{r}(u, v)$ |
| $\frac{\partial \boldsymbol{r}}{\partial v}(u, v)$ | Second tangential vector of the surface parameterized by $\boldsymbol{r}(u, v)$ |
| $\boldsymbol{n}(u, v)$ | Normal of surface parameterized by $\boldsymbol{r}(u, v)$ |
| $\boldsymbol{q_k}$ | Position of the control point $k$ |
| $\phi_k(u, v)$ | Basis function $k$ |
| $N_u$ | Number of control points in the $u$-direction |
| $N_v$ | Number of control points in the $v$-direction |
| $\mathbf{M^0}, \mathbf{M}$ | Metric tensor in reference, and current configurations |
| $[\hat{u}_i]$ | Coordinates of inextensibility grid in the $u$-direction |
| $\left[\hat{v}_j\right]$ | Coordinates of inextensibility grid in the $v$-direction |
| $\hat{N}_u$ | Size of $[\hat{u}_i]$ |
| $\hat{N}_v$ | Size of $\left[\hat{v}_j\right]$ |
| $u_S, v_S$ | Curvilinear coordinates of sensor $S$ |
| $N_S$ | Total number of sensors |
| $\alpha_S$ | Angle around the first tangent vector at the location of sensor $S$ |
| $\beta_S$ | Angle around the second tangent vector at the location of sensor $S$ |

## 2.2 Surface Model

### Definition of Parametric Surface

A 3D surface can be described explicitly by the mapping $\boldsymbol{r} : X \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$. Only two *curvilinear* coordinates $(u, v)$ are needed to uniquely define a point on the surface as shown in figure 2.2. The image of this two-coordinate point through the mapping represents the location of that point in 3D space.

For instance, a flat surface (or plane) can be represented by:

$$\boldsymbol{r}(u, v) = \begin{bmatrix} u \\ v \\ 0 \end{bmatrix} \tag{2.1}$$

A cylindrical surface of radius R along the z-axis can be represented by:

$$\boldsymbol{r}(u, v) = \begin{bmatrix} R\cos u \\ R\sin u \\ v \end{bmatrix} \tag{2.2}$$

### Basis Function Decomposition

We consider a finite dimension mapping defined by basis functions. The mapping $\boldsymbol{r}$ can be written as:

$$\begin{aligned} \boldsymbol{r} : X \subset \mathbb{R}^2 &\longrightarrow \mathbb{R}^3 \\ (u, v) \quad &\longmapsto \boldsymbol{r}(u, v) = \sum_{k=1}^{N} \boldsymbol{q_k} \phi_k(u, v) \end{aligned} \tag{2.3}$$

where $\phi_k : X \rightarrow \mathbb{R}$ are basis functions, $\boldsymbol{q_k}$ are unknown 3D points called control points and define the weight of the basis functions, and $N$ is the dimension of the function space.

This basis representation is common for such problems [54, 59, 73, 86]. Many sets of functions can be used to describe the mapping $\boldsymbol{r}$. Perriollat et al. [59] use Thin-Plate Splines, Metaxas et al. [54] use Finite Element basis functions which are piecewise polynomials defined over local supports. B-Splines are used to fit a surface to data points [24]. Note that B-Splines and the more complicated 2D Non-Uniform Rational Basis Splines (NURBS) are often used in computer-aided design to draw complex surfaces. Other basis functions such as rational Gaussian functions can also be used [34]. They have the advantage of being able to capture

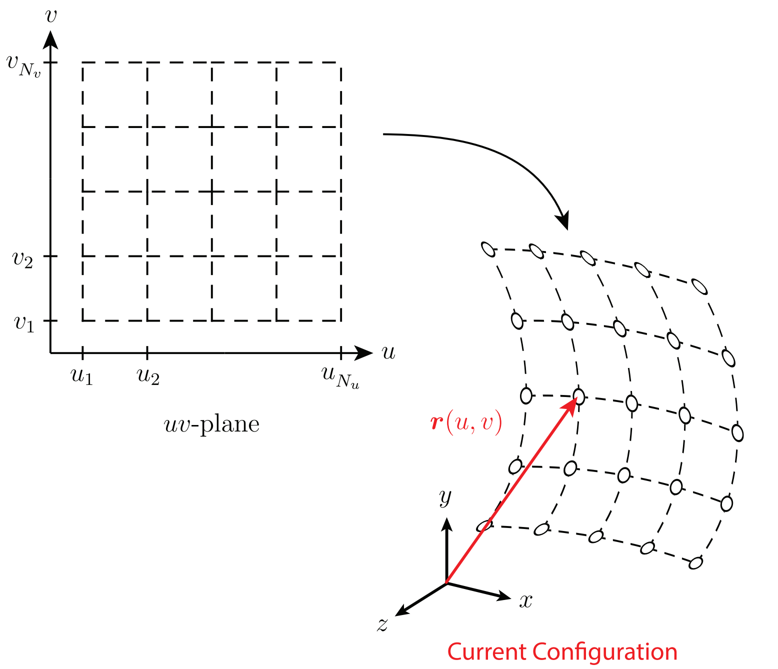


Figure 2.3: Parameterized mapping defining the surface to be reconstructed. Circles represent the 3D position of the control points.

both global and local deformations with one set of basis functions by varying the standard deviation of each Gaussian. Simple polynomial series have also been used in [10].

In order to define angles of the surface at specific locations (see Section 2.4), the basis functions used in our problem need to be differentiable. We limit this study to simple polynomial basis functions: 2D Lagrange polynomials. They are defined over a grid of control points aligned with the curvilinear coordinates (see figure 2.3). Let $N_u$ and $N_v$ be the size of the grid in each direction and note that $N = N_u \times N_v$. We can rewrite equation 2.3 as:

$$\boldsymbol{r}(u,v) = \sum_{k=1}^{N_u} \sum_{l=1}^{N_v} \boldsymbol{q_{k,l}} \phi_{k,l}(u,v) \tag{2.4}$$

Lagrange polynomials are often used to interpolate functions based on known values at discrete locations [2]. They are easy to compute and physically understandable as the control points lie on the surface to be reconstructed. Unfortunately, for a large number of control points, which corresponds to a large polynomial order, they are susceptible to Runge's phenomenon where a function can have large oscillations near the boundaries of the domain [27]. The basis functions are written as:

$$\phi_{k,l}(u,v) = L_k^u(u) L_l^v(v) \tag{2.5}$$

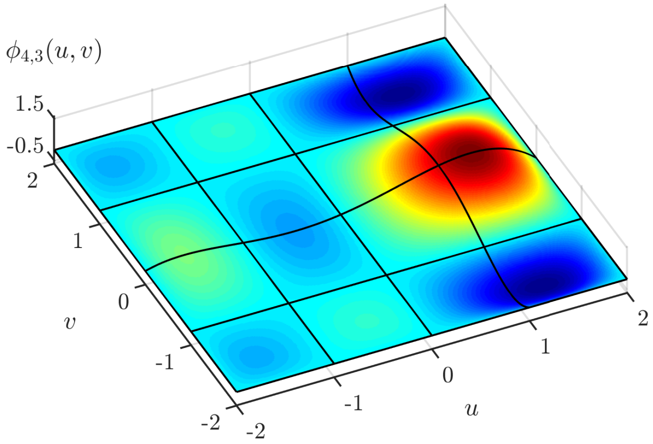


Figure 2.4: Lagrange basis function $\phi_{4,3}(u,v) = L_4^u(u)L_3^v(v)$ on a regular grid of interpolation coordinates $[\boldsymbol{u_k}] = [\boldsymbol{v_l}] = (-2,-1,0,1,2)$.

where $L_k^u$ (resp. $L_l^v$) is the Lagrange polynomial in the $u$-direction (resp. $v$-direction):

$$L_k^u(u) = \prod_{\substack{p=1\\p\neq k}}^{N_u} \frac{u-u_p}{u_k-u_p} \quad \text{and} \quad L_l^v(v) = \prod_{\substack{p=1\\p\neq l}}^{N_v} \frac{v-v_p}{v_l-v_p} \tag{2.6}$$

where $u_k$ (resp. $v_l$) are interpolation coordinates defined in the $uv$-plane. Figure 2.4 shows the Lagrange basis function $\phi_{4,3}(u,v) = L_4^u(u)L_3^v(v)$ on a uniform grid of interpolation coordinates $[\boldsymbol{u_k}] = [\boldsymbol{v_l}] = (-2,-1,0,1,2)$.

## 2.3 Inextensibility Constraints

### Definition of the constraints

Curvilinear distances on a surface in 3D Euclidian space can be calculated using the metric tensor [47, 66]. It is the tensor representation of the first fundamental form in differential geometry [1]. For the parametric surface defined in equation 2.4, the associated metric tensor is:

$$\mathbf{M}(u,v) = \begin{bmatrix} \frac{\partial \boldsymbol{r}}{\partial u} \cdot \frac{\partial \boldsymbol{r}}{\partial u} & \frac{\partial \boldsymbol{r}}{\partial u} \cdot \frac{\partial \boldsymbol{r}}{\partial v} \\ \\ \frac{\partial \boldsymbol{r}}{\partial u} \cdot \frac{\partial \boldsymbol{r}}{\partial v} & \frac{\partial \boldsymbol{r}}{\partial v} \cdot \frac{\partial \boldsymbol{r}}{\partial v} \end{bmatrix} \tag{2.7}$$

where $\boldsymbol{a} \cdot \boldsymbol{b}$ is the inner product between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

The length of a curve defined in the $uv$-space by $(u(t), v(t))$, $t \in [t_0, t_1]$ can be

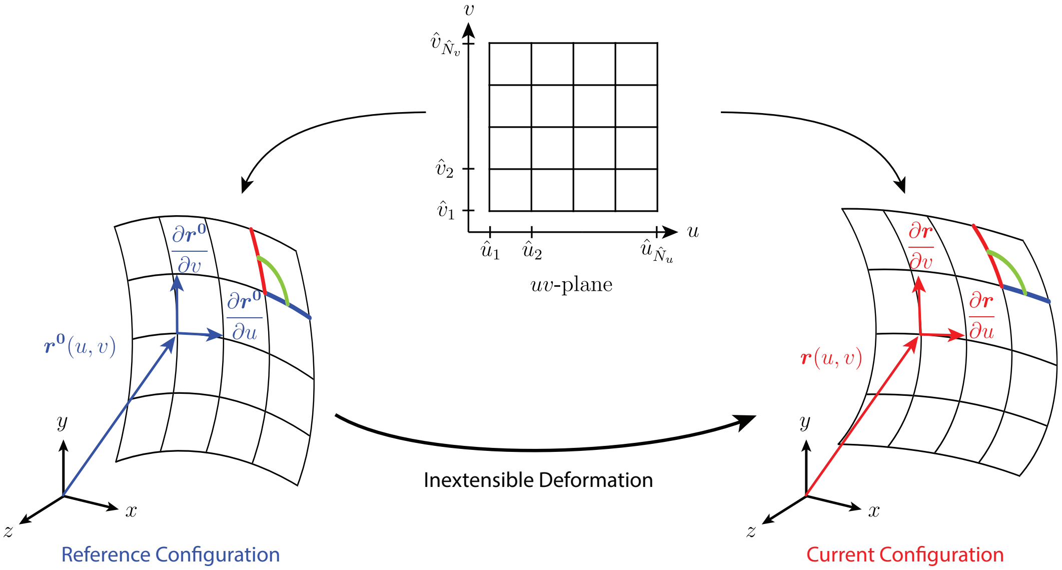


Figure 2.5: Inextensibility constraints on the grid defined by $[\hat{\boldsymbol{u}}_i]$ and $\left[\hat{\boldsymbol{v}}_j\right]$. The red and blue segments remain the same length after deformation. The angle around the node defined by the green arc also remains the same.

calculated as:

$$s = \int_{t_0}^{t_1} \sqrt{\left[u'(t), v'(t)\right] \mathbf{M}(u(t), v(t)) \begin{bmatrix} u'(t) \\ v'(t) \end{bmatrix}}\, dt \tag{2.8}$$

As a result, the deformation between two surfaces defined by $\boldsymbol{r}^{\mathbf{0}}$ and $\boldsymbol{r}$ is inextensible if any curve has the same length on both surfaces, or if and only if the metric tensor is conserved upon deformation $\mathbf{M} = \mathbf{M}^{\mathbf{0}}$:

$$\begin{bmatrix} \frac{\partial \boldsymbol{r}}{\partial u} \cdot \frac{\partial \boldsymbol{r}}{\partial u} & \frac{\partial \boldsymbol{r}}{\partial u} \cdot \frac{\partial \boldsymbol{r}}{\partial v} \\ \frac{\partial \boldsymbol{r}}{\partial u} \cdot \frac{\partial \boldsymbol{r}}{\partial v} & \frac{\partial \boldsymbol{r}}{\partial v} \cdot \frac{\partial \boldsymbol{r}}{\partial v} \end{bmatrix} = \begin{bmatrix} \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial u} \cdot \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial u} & \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial u} \cdot \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial v} \\ \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial u} \cdot \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial v} & \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial v} \cdot \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial v} \end{bmatrix} \tag{2.9}$$

Note that $\mathbf{M}^{\mathbf{0}}$ is fully known since the reference configuration is known. The inextensibility of the transformation leads to the following 3 equations:

$$\frac{\partial \boldsymbol{r}}{\partial u} \cdot \frac{\partial \boldsymbol{r}}{\partial u} = \left\| \frac{\partial \boldsymbol{r}}{\partial u} \right\|^2 = \left\| \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial u} \right\|^2 \tag{2.10}$$

$$\frac{\partial \boldsymbol{r}}{\partial v} \cdot \frac{\partial \boldsymbol{r}}{\partial v} = \left\| \frac{\partial \boldsymbol{r}}{\partial v} \right\|^2 = \left\| \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial v} \right\|^2 \tag{2.11}$$

$$\frac{\partial \boldsymbol{r}}{\partial u} \cdot \frac{\partial \boldsymbol{r}}{\partial v} = \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial u} \cdot \frac{\partial \boldsymbol{r}^{\mathbf{0}}}{\partial v} \tag{2.12}$$

Equations 2.10 and 2.11 impose the condition that the square of the local strain in both directions $u$ and $v$ is zero. Equation 2.12 imposes the condition that the angle between the two tangent directions remains constant.

Equations 2.10, 2.11, and 2.12 form a system of three non-linear differential equations for three unknowns (the components of $\boldsymbol{r}$). The family of solutions represents all possible inextensible deformations from the initial configuration $\boldsymbol{r^0}$. The complexity of this system makes it impossible to solve analytically. As a result, we derive a finite subset of constraints inspired from these PDEs.

We define a regular grid called *Inextensibility grid* aligned with the curvilinear coordinates $u$ and $v$. It is parameterized by the coordinates $[\boldsymbol{\hat{u}_i}] = (\hat{u}_1, ..., \hat{u}_i, ..., \hat{u}_{\hat{N}_u})$ and $\left[\boldsymbol{\hat{v}_j}\right] = (\hat{v}_1, ..., \hat{v}_j, ..., \hat{v}_{\hat{N}_v})$. This grid is shown in Figure 2.5. It is usually different from the grid used to define the Lagrange polynomials shown in Figure 2.3.

Equations 2.10, 2.11, and 2.12 are discretized into a finite number of constraints on this grid. The strain constraints are applied on average between two nodes of the grid. This is equivalent to conserving the length of each edge of the grid. The dot product constraint is taken at the nodes of the grid. This leads to the following equations:

$$\int_{\hat{u}_i}^{\hat{u}_{i+1}} \left\| \frac{\partial \boldsymbol{r}}{\partial u}(u, \hat{v}_j) \right\| du = \int_{\hat{u}_i}^{\hat{u}_{i+1}} \left\| \frac{\partial \boldsymbol{r^0}}{\partial u}(u, \hat{v}_j) \right\| du \tag{2.13}$$

$$\int_{\hat{v}_j}^{\hat{v}_{j+1}} \left\| \frac{\partial \boldsymbol{r}}{\partial v}(\hat{u}_i, v) \right\| dv = \int_{\hat{v}_j}^{\hat{v}_{j+1}} \left\| \frac{\partial \boldsymbol{r^0}}{\partial v}(\hat{u}_i, v) \right\| dv \tag{2.14}$$

$$\frac{\partial \boldsymbol{r}}{\partial u}^T (\hat{u}_i, \hat{v}_j) \frac{\partial \boldsymbol{r}}{\partial v}(\hat{u}_i, \hat{v}_j) = \frac{\partial \boldsymbol{r^0}}{\partial u}^T (\hat{u}_i, \hat{v}_j) \frac{\partial \boldsymbol{r^0}}{\partial v}(\hat{u}_i, \hat{v}_j) \tag{2.15}$$

where the right hand-side is known.

These constraint equations can be physically interpreted by considering a 2D structure made of rods and rigid joints. The length constraints represent the inextensibility of the rods while the angle constraints represent the fixed angles between rods imposed by the joints.

A similar approach is used in [15]. The metric tensor is constrained to be equal to the identity tensor at discrete locations on the surface (nodes of a grid). The length constraints in equations 2.13 and 2.14 are replaced by simply imposing the norm of the tangent vectors to be 1 at these nodes while their dot product is equal to 0. This approach limits the problem to developable surfaces where $u$ and $v$ are Cartesian coordinates of a point in the flat configuration. Our methodology can be applied to any kind of surface, developable or not, and using lengths in the constraints as opposed to the norm of the tangent vectors makes it easier to define the reference configuration.

### Rank Deficiencies

The inextensibility conditions defined in equations 2.13, 2.14, and 2.15 applied over the whole inextensibility grid define a system of $3\hat{N}_u\hat{N}_v - \hat{N}_u - \hat{N}_v$ equations

$$\boldsymbol{f}(\boldsymbol{q_1}, ..., \boldsymbol{q_N}) = 0 \tag{2.16}$$

where $\boldsymbol{q_k}$ is defined in equation 2.3, $\boldsymbol{f}$ is a vector containing all the inextensibility conditions, written as:

$$\begin{aligned} f_{u,i,j} = & \int_{\hat{u}_i}^{\hat{u}_{i+1}} \left\| \sum_{k=1}^{N} \boldsymbol{q_k} \frac{\partial \phi_k}{\partial u}(u, \hat{v}_j) \right\| du \\ & - \int_{\hat{u}_i}^{\hat{u}_{i+1}} \left\| \frac{\partial \boldsymbol{r^0}}{\partial u}(u, \hat{v}_j) \right\| du = 0 \end{aligned} \tag{2.17}$$

$$\begin{aligned} f_{v,i,j} = & \int_{\hat{v}_j}^{\hat{v}_{j+1}} \left\| \sum_{k=1}^{N} \boldsymbol{q_k} \frac{\partial \phi_k}{\partial v}(\hat{u}_i, v) \right\| dv \\ & - \int_{\hat{v}_j}^{\hat{v}_{j+1}} \left\| \frac{\partial \boldsymbol{r^0}}{\partial v}(\hat{u}_i, v) \right\| dv = 0 \end{aligned} \tag{2.18}$$

$$\begin{aligned} f_{A,i,j} = & \sum_{k=1}^{N} \sum_{l=1}^{N} \boldsymbol{q_k}^T \boldsymbol{q_l} \frac{\partial \phi_k}{\partial u}(\hat{u}_i, \hat{v}_j) \frac{\partial \phi_l}{\partial v}(\hat{u}_i, \hat{v}_j) \\ & - \frac{\partial \boldsymbol{r^0}}{\partial u}^T (\hat{u}_i, \hat{v}_j) \frac{\partial \boldsymbol{r^0}}{\partial v}(\hat{u}_i, \hat{v}_j) = 0 \end{aligned} \tag{2.19}$$

The Jacobian of this system of equations is defined as the tensor:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q_1}} & \dots & \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q_N}} \end{bmatrix} \tag{2.20}$$

Let $(\boldsymbol{q_1^*}, ..., \boldsymbol{q_N^*})$ be a solution of equation 2.16 and $\boldsymbol{r^*}$ the associated surface. Let $\boldsymbol{q_k} = \boldsymbol{q_k^*} + \boldsymbol{\delta q_k}$ with $\|\boldsymbol{\delta q_k}\| \ll 1$. One can calculate $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q_k}}$ by calculating the first order term of the Taylor expansion of $\boldsymbol{f}(\boldsymbol{q_1^*}, ..., \boldsymbol{q_k^*} + \boldsymbol{\delta q_k^*}, ..., \boldsymbol{q_N})$. This leads to:

$$\frac{\partial f_{u,i,j}}{\partial \boldsymbol{q_k}} = \int_{\hat{u}_i}^{\hat{u}_{i+1}} \frac{\frac{\partial \boldsymbol{r^*}}{\partial u}(u, \hat{v}_j)}{\left\| \frac{\partial \boldsymbol{r^*}}{\partial u}(u, \hat{v}_j) \right\|} \frac{\partial \phi_k}{\partial u}(u, \hat{v}_j) du \tag{2.21}$$

$$\frac{\partial f_{v,i,j}}{\partial \boldsymbol{q_k}} = \int_{\hat{v}_i}^{\hat{v}_{i+1}} \frac{\frac{\partial \boldsymbol{r^*}}{\partial v}(\hat{u}_i, v)}{\left\| \frac{\partial \boldsymbol{r^*}}{\partial v}(\hat{u}_i, v) \right\|} \frac{\partial \phi_k}{\partial v}(\hat{u}_i, v) dv \tag{2.22}$$

$$\frac{\partial f_{A,i,j}}{\partial \boldsymbol{q_k}} = \left( \frac{\partial \phi_k}{\partial v} \frac{\partial \boldsymbol{r^*}}{\partial u} + \frac{\partial \phi_k}{\partial u} \frac{\partial \boldsymbol{r^*}}{\partial v} \right) (\hat{u}_i, \hat{v}_j) \tag{2.23}$$

Singularities of the Jacobian (deformations associated with a null singular value) describe the possible inextensible motions of the surface around the current configuration.

It is easy to show that any rigid-body translations and infinitesimal rigid-body rotations are singularities of the Jacobian (Equations 2.21, 2.22, and 2.23 are equal to 0).

One can also show that because equations 2.21, 2.22, and 2.23 are functions of the basis function derivatives in $u$ and $v$, the inextensibility grid has to cover the whole surface. The value of the basis function derivatives decreases as the distance from a control point increases. If the grid is localized in a certain region, the motion of a control point far from this region can create a numerical singularity as equations 2.21, 2.22, and 2.23 become close to 0, hence creating a near-zero column in the Jacobian matrix.

Additionally to rigid-body motions, it is possible to have non-rigid singularities. These are important since they correspond to actual deformations of the surface.

We decide to study the case of an initially flat surface. It is interesting since it has no curvature, which results in no preferred motion. Without loss of generality, we assume that the surface is perpendicular to the z-axis. The last coordinate of $\boldsymbol{r^*}$ is constant over the surface. One can show that the last coordinate of $\frac{\partial \boldsymbol{r^*}}{\partial u}$ and $\frac{\partial \boldsymbol{r^*}}{\partial v}$ is 0. As a result, $\frac{\partial f_{u,i,j}}{\partial \boldsymbol{q_k}}$, $\frac{\partial f_{v,i,j}}{\partial \boldsymbol{q_k}}$, and $\frac{\partial f_{A,i,j}}{\partial \boldsymbol{q_k}}$ are normal to the z-axis and any vector $\boldsymbol{\delta q_k}$ along the z-axis leads to a singularity.

It is easy to prove that such a surface has a total of $N_u N_v + 3$ singularities. This corresponds to the local singularity of each control point (movement along z) plus the rigid translations in x and y and the rigid rotation around z. One can also show that this represents the maximum number of singular deformations. Any curved surface has fewer singularities because the non-zero curvatures force the surface to deform in certain directions only.

Figure 2.6 shows a singular motion for an initially cylindrical shape using Lagrange polynomials as defined in figure 2.4 ($5 \times 5$ uniform grid of control points). Since the polynomials cannot exactly reconstruct a cylinder, the initial shape is not exactly a cylinder, but the result of constraining each control point to its position on the perfect cylinder (see equation 2.41). This singularity was obtained numerically using

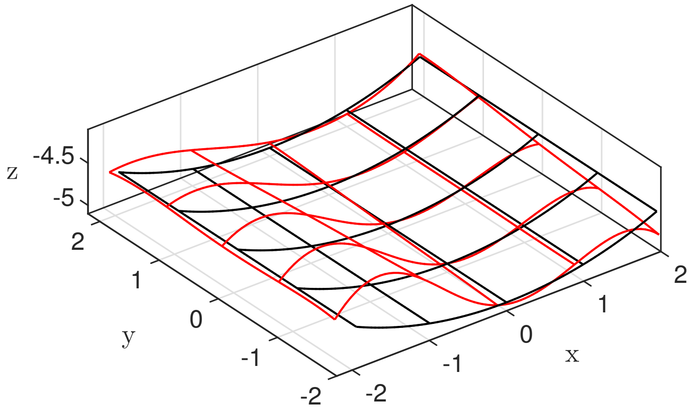


Figure 2.6: Numerical singularity of the inextensibility constraints for a cylinder of radius 5 using 5 × 5 regularly spaced control points on an inextensibility grid $[\hat{\boldsymbol{u}}_i] = [\hat{\boldsymbol{v}}_j] = (-2, -1, 0, 1, 2)$. The black shape represents the initial cylinder and the red shape represents the added singularity.

a Singular Value Decomposition (SVD) of the Jacobian matrix in MATLAB. A total of 15 singularities were found (6 rigid-body motions and 9 non-rigid deformations). Identical grids on an initially flat shape led to 28 singularities.

## 2.4 Angle Measurement Constraints

### Definition of the measurement

In order to measure the deformation of the surface, we introduce local angle measurements at several points on the surface. These measurements determine the angles between the normal to the surface at discrete points and a specific direction.

We assume that the angles are measured to the line from the sensor location and the origin of $\mathbb{R}^3$ which coincides with the vector $\boldsymbol{r}$. This can be done in practice by placing a light source at the origin and light sensors on the surface.

At each sensor location, two angles $\alpha, \beta$ are measured along the curvilinear coordinates, that is along $\frac{\partial \boldsymbol{r}}{\partial u}$ and $\frac{\partial \boldsymbol{r}}{\partial v}$ from the normal of the surface as, shown in figure 2.7.

The location of the sensors is defined by $(u_S, v_S) \in \mathbb{R}^{N_S \times 2}$ where $N_S$ is the total number of sensors. They do not need to lie on a specific grid as required by previous research [38, 80]. Figure 2.8 shows an example of the location of the sensors in the $uv$-space and their position in the current configuration. The local coordinate system at the location of a sensor is also shown.

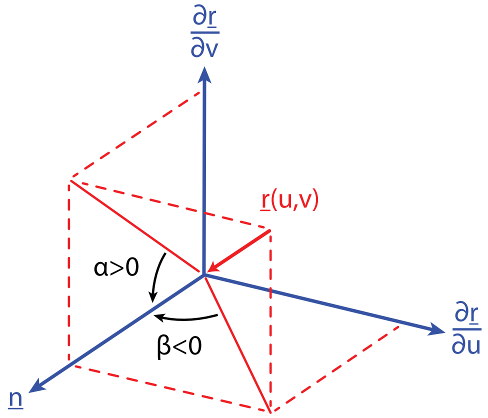


Figure 2.7: Definition of the angles in the local coordinate system of a sensor. $\boldsymbol{r}(u_S, v_S)$ represents the direction to which the angles are measured. Note that the angle $\beta$ is negative in the figure (positive angles are defined from $\boldsymbol{n}$ to $\frac{\partial \boldsymbol{r}}{\partial u}$).

The angles at a sensor location $(u_S, v_S)$ are defined as:

$$\tan \alpha_S = \frac{\boldsymbol{r}(u_S, v_S) \cdot \frac{\partial \boldsymbol{r}}{\partial v}(u_S, v_S)}{\boldsymbol{r}(u_S, v_S) \cdot \boldsymbol{n}(u_S, v_S)} \frac{\|\boldsymbol{n}(u_S, v_S)\|}{\left\|\frac{\partial \boldsymbol{r}}{\partial v}(u_S, v_S)\right\|} \tag{2.24}$$

$$\tan \beta_S = -\frac{\boldsymbol{r}(u_S, v_S) \cdot \frac{\partial \boldsymbol{r}}{\partial u}(u_S, v_S)}{\boldsymbol{r}(u_S, v_S) \cdot \boldsymbol{n}(u_S, v_S)} \frac{\|\boldsymbol{n}(u_S, v_S)\|}{\left\|\frac{\partial \boldsymbol{r}}{\partial u}(u_S, v_S)\right\|} \tag{2.25}$$

where $\boldsymbol{n}(u_S, v_S)$ is the normal to the surface:

$$\boldsymbol{n} = \frac{\partial \boldsymbol{r}}{\partial u} \times \frac{\partial \boldsymbol{r}}{\partial v} \tag{2.26}$$

Note that similar equations can be written when the angle measurement is taken from a fixed direction such as the gravity vector $\boldsymbol{g}$. Then equations 2.24 and 2.25 become:

$$\tan \alpha_S = \frac{\boldsymbol{g} \cdot \frac{\partial \boldsymbol{r}}{\partial v}(u_S, v_S)}{\boldsymbol{g} \cdot \boldsymbol{n}(u_S, v_S)} \frac{\|\boldsymbol{n}(u_S, v_S)\|}{\left\|\frac{\partial \boldsymbol{r}}{\partial v}(u_S, v_S)\right\|} \tag{2.27}$$

$$\tan \beta_S = -\frac{\boldsymbol{g} \cdot \frac{\partial \boldsymbol{r}}{\partial u}(u_S, v_S)}{\boldsymbol{g} \cdot \boldsymbol{n}(u_S, v_S)} \frac{\|\boldsymbol{n}(u_S, v_S)\|}{\left\|\frac{\partial \boldsymbol{r}}{\partial u}(u_S, v_S)\right\|} \tag{2.28}$$

**Rank Deficiency**

The equations 2.24 and 2.25 are invariant for any rotation around the origin. Uniform scaling (multiplying $\boldsymbol{r}$ by a non-zero coefficient) also creates a singularity.

Numerical rank deficiencies (singular values close to 0) can occur when sensors are concentrated on a specific part of the structure. This has similar effect to localizing the inextensibility grid on one part of the surface (see section 2.3).

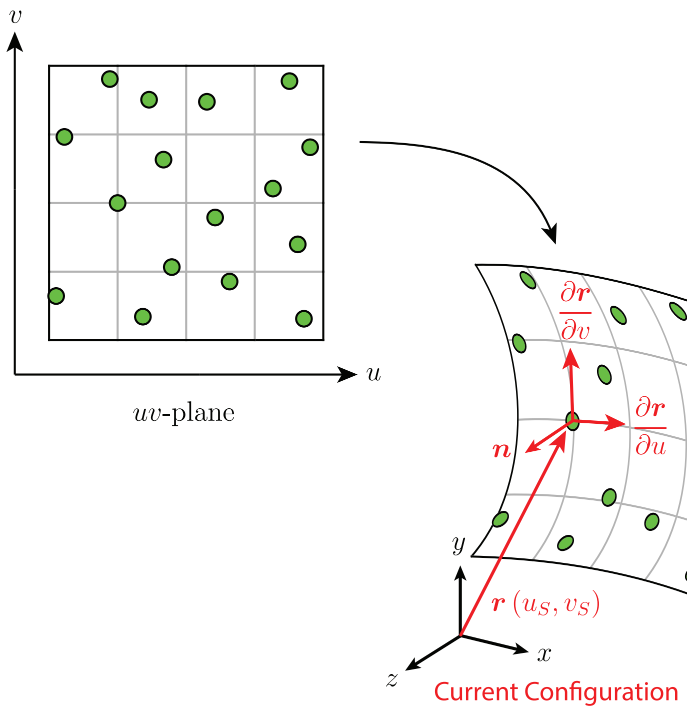

$v$
$u$
$uv$-plane
$\frac{\partial \boldsymbol{r}}{\partial v}$
$\frac{\partial \boldsymbol{r}}{\partial u}$
$\boldsymbol{n}$
$y$
$\boldsymbol{r}\,(u_S, v_S)$
$x$
$z$
Current Configuration

Figure 2.8: Example position of the angle sensors on the surface (green circles). The vector defining the position of a sensor $\boldsymbol{r}(u_S, v_S)$ is collinear with the light ray seen by the sensor (the light source is positioned at the origin). The local coordinate system at that sensor location is also shown.

To show this, we calculate the derivatives of the equations 2.24 and 2.25 with respect to $\boldsymbol{q_k}$. Because of the complexity of these equations, it is assumed that the deformation is perfectly inextensible. This means that $\left\|\frac{\partial \boldsymbol{r}}{\partial u}\right\|$, $\left\|\frac{\partial \boldsymbol{r}}{\partial v}\right\|$, and $\|\boldsymbol{n}\|$ are constant. Without loss of generality, they are set to 1. Additionally, the light source is assumed to be very far from the surface and along the z-direction. The dot products involving $\boldsymbol{r}(u_S, v_S)$ can be replaced by dot products with $\boldsymbol{z}$ (as in equations 2.27 and 2.28 by replacing $\boldsymbol{g}$ with $\boldsymbol{z}$)

With these simplifications, one can show that the derivatives about an initial shape $\boldsymbol{r}^*$ can be written as:

$$\frac{\partial \tan\alpha_S}{\partial \boldsymbol{q_k}} = \frac{\tan\alpha_S^*}{\boldsymbol{z}\cdot\boldsymbol{n}^*}\begin{bmatrix} \frac{\partial \boldsymbol{r}^*}{\partial v}\frac{\partial\phi_k}{\partial u} - \frac{\partial \boldsymbol{r}^*}{\partial u}\frac{\partial\phi_k}{\partial v} \\ \frac{\partial \boldsymbol{r}^*}{\partial u}\frac{\partial\phi_k}{\partial v} - \frac{\partial \boldsymbol{r}^*}{\partial v}\frac{\partial\phi_k}{\partial u} \\ \frac{1}{\tan\alpha_S^*}\frac{\partial\phi_k}{\partial v} \end{bmatrix} \tag{2.29}$$

$$\frac{\partial \tan\beta_S}{\partial \boldsymbol{q_k}} = \frac{\tan\beta_S^*}{\boldsymbol{z}\cdot\boldsymbol{n}^*}\begin{bmatrix} \frac{\partial \boldsymbol{r}^*}{\partial v}\frac{\partial\phi_k}{\partial u} - \frac{\partial \boldsymbol{r}^*}{\partial u}\frac{\partial\phi_k}{\partial v} \\ \frac{\partial \boldsymbol{r}^*}{\partial u}\frac{\partial\phi_k}{\partial v} - \frac{\partial \boldsymbol{r}^*}{\partial v}\frac{\partial\phi_k}{\partial u} \\ \frac{1}{\tan\beta_S^*}\frac{\partial\phi_k}{\partial u} \end{bmatrix} \tag{2.30}$$

From these equations, we can see that if $\left\|\frac{\partial \phi_k}{\partial u}\right\| \ll 1$ and/or $\left\|\frac{\partial \phi_k}{\partial v}\right\| \ll 1$ for all $(u_S, v_S) \in \mathbb{R}^{N_S \times 2}$, then both derivatives have components close to 0 and some columns of the Jacobian will be close to 0 leading to a numerical singularity. This would be the case if, for example, only a part of the surface is covered with angle sensors. The shape functions associated with the control points located far away from the sensors will have near-zero derivatives at the sensor locations.

## 2.5 Surface Reconstruction as a Least-Squares Problem

### System of Equations

Equations 2.13, 2.14, 2.15, 2.24, and 2.25 form a system that solves the problem of reconstructing a surface from angle measurements undergoing an inextensible deformation from a template:

$$
\begin{aligned}
&\int_{\hat{u}_i}^{\hat{u}_{i+1}} \left\|\frac{\partial \boldsymbol{r}}{\partial u}(u, \hat{v}_j)\right\| du - \int_{\hat{u}_i}^{\hat{u}_{i+1}} \left\|\frac{\partial \boldsymbol{r^0}}{\partial u}(u, \hat{v}_j)\right\| du = 0 \\
&\qquad \forall i = 1, \ldots, \hat{N}_u - 1, \quad \forall j = 1, \ldots, \hat{N}_v \\[2ex]
&\int_{\hat{v}_j}^{\hat{v}_{j+1}} \left\|\frac{\partial \boldsymbol{r}}{\partial v}(\hat{u}_i, v)\right\| dv - \int_{\hat{v}_j}^{\hat{v}_{j+1}} \left\|\frac{\partial \boldsymbol{r^0}}{\partial v}(\hat{u}_i, v)\right\| dv = 0 \\
&\qquad \forall i = 1, \ldots, \hat{N}_u, \quad \forall j = 1, \ldots, \hat{N}_v - 1 \\[2ex]
&\frac{\partial \boldsymbol{r}}{\partial u}^T(\hat{u}_i, \hat{v}_j)\frac{\partial \boldsymbol{r}}{\partial v}(\hat{u}_i, \hat{v}_j) = \frac{\partial \boldsymbol{r^0}}{\partial u}^T(\hat{u}_i, \hat{v}_j)\frac{\partial \boldsymbol{r^0}}{\partial v}(\hat{u}_i, \hat{v}_j) \\
&\qquad \forall i = 1, \ldots, \hat{N}_u, \quad \forall j = 1, \ldots, \hat{N}_v \\[2ex]
&\tan\alpha - \frac{\boldsymbol{r}(u_S, v_S) \cdot \frac{\partial \boldsymbol{r}}{\partial v}(u_S, v_S)}{\boldsymbol{r}(u_S, v_S) \cdot \boldsymbol{n}(u_S, v_S)} \frac{\|\boldsymbol{n}(u_S, v_S)\|}{\left\|\frac{\partial \boldsymbol{r}}{\partial v}(u_S, v_S)\right\|} = 0 \\
&\qquad \forall S = 1, \ldots, N_S \\[2ex]
&\tan\beta + \frac{\boldsymbol{r}(u_S, v_S) \cdot \frac{\partial \boldsymbol{r}}{\partial u}(u_S, v_S)}{\boldsymbol{r}(u_S, v_S) \cdot \boldsymbol{n}(u_S, v_S)} \frac{\|\boldsymbol{n}(u_S, v_S)\|}{\left\|\frac{\partial \boldsymbol{r}}{\partial u}(u_S, v_S)\right\|} = 0 \\
&\qquad \forall S = 1, \ldots, N_S
\end{aligned}
\tag{2.31}
$$

Replacing $\boldsymbol{r}$ with equation 2.3 everywhere, the system is a function of the unknown control point coordinates, and can be written as:

$$f_k(\bar{\boldsymbol{q}}) = 0 \qquad \forall k = 1, ..., N_{eq} \tag{2.32}$$

where $f_k(\cdot)$ represents the equations defined in 2.31, $N_{eq} = 3\hat{N}_u\hat{N}_v - \hat{N}_u - \hat{N}_v + 2N_S$ is the number of equations in the system, and $\bar{\boldsymbol{q}}$ is the unknown vector defined as the vertical concatenation of the control points. Since this system is (usually) overconstrained, we rewrite the problem as a least-squares minimization:

$$\bar{\boldsymbol{q}}^* = \arg\min_{\bar{q}} \sum_{k=1}^{N_{eq}} f_k^2(\bar{\boldsymbol{q}}) = \arg\min_{\bar{q}} \|\boldsymbol{f}(\bar{\boldsymbol{q}})\|^2 \tag{2.33}$$

where $\boldsymbol{f}(\bar{\boldsymbol{q}}) = [f_1(\bar{\boldsymbol{q}}), ..., f_{N_{eq}}(\bar{\boldsymbol{q}})]^T$.

**Solution of Overconstrained System**

In order to solve the least-squares problem defined in equation 2.33, we use the Levenberg-Marquardt algorithm in MATLAB [26].

This iterative algorithm starts with an initial guess $\bar{\boldsymbol{q}}_{\mathbf{0}}$, finds increments of the unknown vector, $\boldsymbol{\delta}\bar{\boldsymbol{q}}_{\boldsymbol{k}}$, such that at the next step, $\bar{\boldsymbol{q}}_{\boldsymbol{k+1}} = \bar{\boldsymbol{q}}_{\boldsymbol{k}} + \boldsymbol{\delta}\bar{\boldsymbol{q}}_{\boldsymbol{k}}$. The increment at each step solves the equation:

$$\left(\mathbf{J}^T\mathbf{J} + \lambda_k\mathbf{I}\right) \boldsymbol{\delta}\bar{\boldsymbol{q}}_{\boldsymbol{k}} = -\mathbf{J}^T\boldsymbol{f}(\bar{\boldsymbol{q}}_{\boldsymbol{k}}) \tag{2.34}$$

where $\mathbf{J} = \frac{\partial f}{\partial \bar{q}}$ is the Jacobian of the system and $\lambda_k$ is a non-negative damping factor that is optimized at each step to maximize the decrease in the residual. If $\lambda_k = 0$, this algorithm is equivalent to the Gauss-Newton algorithm (GNA). It is, however, more robust than GNA when the initial guess is far from the solution. When $\lambda_k \longrightarrow \infty$, the algorithm tends to the gradient descent algorithm. As a result, the Levenberg-Marquardt method is a hybrid of the two algorithms.

The Levenberg-Marquardt algorithm stops when the increment is smaller than a prescribed tolerance , i.e. when:

$$\|\boldsymbol{\delta}\bar{\boldsymbol{q}}_{\boldsymbol{k}}\| < \epsilon \tag{2.35}$$

It is important to note that since the system is over-constrained and the stopping criterion involves the norm of the increment, the scaling of the equations 2.31 matters. This will be shown through examples in the next section.

### Rank Deficiencies

The Jacobian of the system can be calculated as the vertical concatenation of the Jacobian studied in section 2.3 and 2.4. In order to use the Levenberg-Marquardt algorithm, the Jacobian needs to be full rank, i.e., its columns need to be independent.

A first requirement is to have more equations than unknowns: $N_{eq} = 3\hat{N}_u\hat{N}_v - \hat{N}_u - \hat{N}_v + 2N_S > 3N_uN_v$.

From the previous observations in sections 2.3 and 2.4, all equations are invariant for rigid-body rotations around the origin. Without loss of generality, we constrain 3 coordinates among the control points to be fixed. For instance, one point is restrained from moving along the x-axis and y-axis, and another point is constrained from moving along the x-axis.

Additionally, the inextensibility conditions can have up to $N = N_uN_v$ singularities as explained in section 2.3. These are mutually exclusive with the singularities of the angle equations. As a result, a minimum of $N$ angle measurements are needed to have a full rank Jacobian, i.e. at least as many angle measurements as control points.

It is important to notice that when the distance from the origin to the surface becomes large, another numerical singularity emerges. It corresponds to a spherical motion. Figure 2.9 shows a schematic of this singularity in 2D. A line perpendicular to the light source can conform to any circle, hence there are infinitely many solutions. In 3D, this is not a singularity as its associated singular value is not exactly 0: conforming a section of a sphere to a sphere of different radius is an extensible deformation. This singularity is however orders of magnitude lower than the next higher which causes numerical issues. We call this singularity a *spherical singularity*.

To remedy this issue, the distance of a point on the surface to the origin can be fixed or bounded.

In the case where the direction of the angle measurement is fixed (equations 2.27 and 2.28), this numerical singularity becomes an actual singularity corresponding to the rigid-body translation along that direction. Fixing a point in 3D space can be done without loss of generality.

## 2.6 Applications to Developable Surfaces

Developable surfaces are simple to reconstruct as they can be mapped onto a plane where the reference configuration can easily be defined. Such shapes include

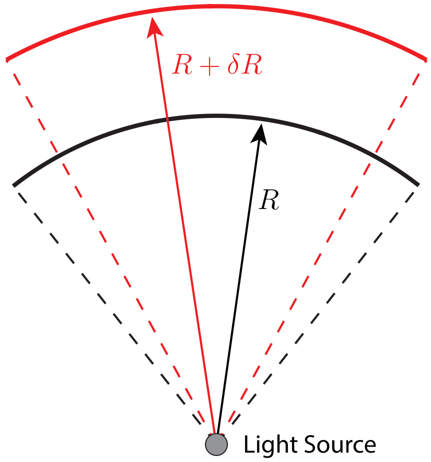


Figure 2.9: 2D representation of the spherical singularity of the system of equations. The black and red lines have the same length and angles to the light source, but their shapes are different.

cylinders, cones, tangent developable surfaces, etc.

We apply the presented method to reconstruct a cylinder and a cone from an initially flat surface. The initial surface is described by:

$$\boldsymbol{r}^{\mathbf{0}}(u,v) = \begin{bmatrix} u \\ v \\ z \end{bmatrix} \tag{2.36}$$

where z is a non-zero constant.

The surface then deforms inextensibly into a cylinder or cone. The measured angles are extracted from the analytical expressions of the cylinder or cone $\tilde{\boldsymbol{r}}$ using equations 2.24 and 2.25. These angles are used as inputs to the algorithm, together with the inextensibility constraints, and 3 coordinates are fixed to prevent rigid-body rotations, as described in section 2.5.

The accuracy of the method will be evaluated for the two test cases (cylinder and cone). The error is calculated as the norm of the vector joining the points with equal coordinates $(u,v)$ in the reconstructed and exact shapes:

$$\text{Error}(u,v) = \|\boldsymbol{r}(u,v) - \tilde{\boldsymbol{r}}(u,v)\| \tag{2.37}$$

The inextensibility of the surface deformation is also evaluated by plotting the error of the 3 elements of the metric tensor from their nominal values defined in equations 2.10, 2.11, and 2.12. Note that in this case, $\mathbf{M}^{\mathbf{0}} = \mathbf{I}$, the $2 \times 2$ identity tensor.

To better understand the elements of the tensor, we plot the Lagrangian normal and shear strains of the surface [52]. They are given by:

$$\epsilon_u = \frac{1}{2}\left(\mathbf{M_{11}} - 1\right) \tag{2.38}$$

$$\epsilon_v = \frac{1}{2}\left(\mathbf{M_{22}} - 1\right) \tag{2.39}$$

$$\gamma_{uv} = \mathbf{M_{12}} \tag{2.40}$$

The convergence of the method when increasing the number of basis functions is shown. The different parameters defining the algorithm are varied and studied. We show the existence of optimal parameters that minimize the error of the reconstruction.

### Reconstruction of a Cylinder

The equation of the inextensible transformation of a flat plane onto a cylinder of radius R aligned with the y-axis is:

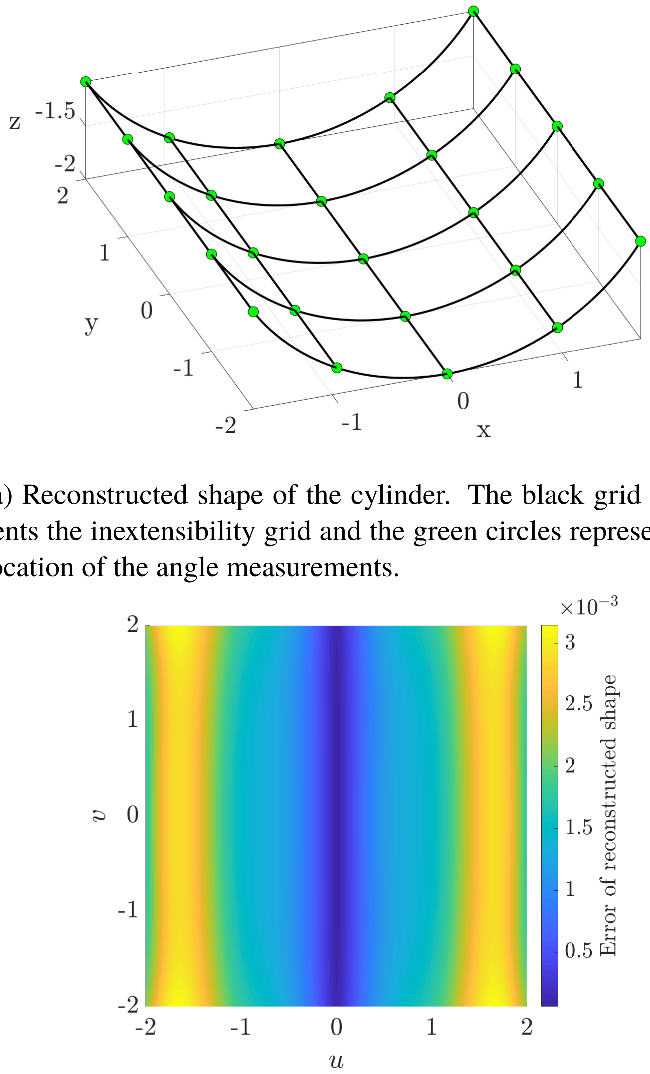
$$\tilde{\boldsymbol{r}}(u,v) = \begin{bmatrix} R\sin\left(\frac{u}{R}\right) \\ v \\ -R\cos\left(\frac{u}{R}\right) \end{bmatrix} \tag{2.41}$$

The local angles associated with this shape are:

$$\tan\alpha_S = -\frac{v_S}{R} \tag{2.42}$$

$$\tan\beta_S = 0 \tag{2.43}$$

These angles were evaluated at the 25 points shown in figure 2.10a and their values were fed to the algorithm in order to reconstruct the cylinder. The reconstructed shape was defined by Lagrange polynomials built from a uniform grid of interpolation points $[\boldsymbol{u_k}] = [\boldsymbol{v_l}] = (-2,-1,0,1,2)$. Figure 2.10a shows the reconstructed cylinder based on a 5 × 5 uniform inextensibility grid $[\boldsymbol{\hat{u}_i}] = \left[\boldsymbol{\hat{v}_j}\right] = (-2,-1,0,1,2)$ and angle measurements (shown by green circles) placed at the nodes of the same grid. This ensures that the number of angle constraints is greater than the number of control points to avoid the singularities shown in section 2.3. The central point (of curvilinear coordinates $(u,v) = (0,0)$) was constrained along the z-axis (x and y coordinates set to 0) and the x-coordinate of the point of curvilinear coordinate



(a) Reconstructed shape of the cylinder. The black grid represents the inextensibility grid and the green circles represent the location of the angle measurements.

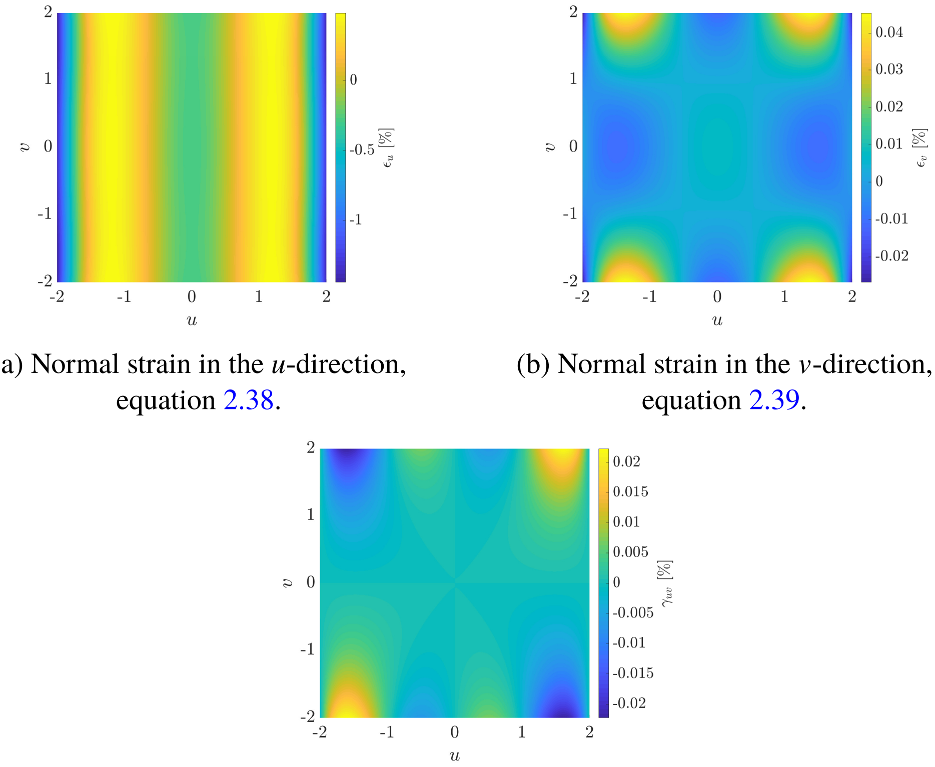
(b) Error between reconstructed cylinder shown in figure 2.10a and cylinder defined by equation 2.41.

Figure 2.10: Reconstruction of a cylinder of radius 2 aligned with the y-axis using Lagrange polynomials on uniform grids $[\boldsymbol{u_k}] = [\boldsymbol{v_l}] = [\boldsymbol{\hat{u}_i}] = \left[\boldsymbol{\hat{v}_j}\right] = (-2, -1, 0, 1, 2)$.

$(u, v) = (0, 1)$ was constrained to $x = 0$. The initial shape of the algorithm was a flat plane perpendicular to the z-axis:

$$\boldsymbol{r^0}(u, v) = \begin{bmatrix} u \\ v \\ -2 \end{bmatrix} \tag{2.44}$$

It was found that the algorithm converges if this initial plane is close enough to the



(a) Normal strain in the $u$-direction, equation 2.38.

(b) Normal strain in the $v$-direction, equation 2.39.

(c) Shear strain, equation 2.40.

Figure 2.11: Strains of the reconstructed cylinder of radius 2 aligned with the y-axis using Lagrange polynomials on uniform grids $[\boldsymbol{u_k}] = [\boldsymbol{v_l}] = [\boldsymbol{\hat{u}_i}] = \left[\boldsymbol{\hat{v}_j}\right] = (-2, -1, 0, 1, 2)$.

solution. If it is too far, the spherical singularity discussed in section 2.5 will force the shape to translate far away along the $-z$ direction.

Figure 2.10b shows the error between the reconstructed shape and the analytical cylinder. We see that the algorithm reconstructs the overall shape of the cylinder with great accuracy. The error is independent in $v$ since there is no deformation in this direction. It is close to 0 at the center of the reconstructed shape where the two points constraining rigid-body rotations are located (and fixed to their exact coordinates) and increases towards the edge of the surface.

Figure 2.11 shows the strain components across the surface. The normal strains are close to 0 on average along each segment of the inextensibility grid. However, they still vary across the surface. A similar observation can be made for the shear strains that are also close to 0 at each node of the inextensibility grid, but are non-zero elsewhere.

The normal strain in the $u$-direction is dominant and mostly varies along $u$. Since

Lagrange polynomials cannot perfectly reconstruct a circle, the algorithm needs to stretch the surface in order for the measured angles to better match their set values. More precisely, the algorithm does a tradeoff between slightly stretching the surface, and adding some bias to the measured angles.

The shear strain variations are related to the normal strains: the small non-uniform accumulated strain displaces the points of the surface forcing it to shear. Figure 2.11b shows that the maximum strain occurs along the outer edges in the $v$-direction which is where the shear is also maximum.

### Reconstruction of a Cone

As a next step, we reconstruct a more complex developable shape. A cone is a more demanding test because a flat surface has to deform in both directions to conform to a conical shape [47]. The equation of a cone around the y-axis is:

$$\tilde{\boldsymbol{r}}(u,v) = \begin{bmatrix} [H - t(u,v)]\sin\gamma\sin\theta(u,v) \\ t(u,v)\cos\gamma \\ [H - t(u,v)]\sin\gamma\cos\theta(u,v) \end{bmatrix} \tag{2.45}$$

where $\theta$ is the circumferential angle around the y-axis, $H$ is the distance from the 'center' of the surface to the tip of the cone, $t$ is the additional distance of a point from $H$, and $\gamma$ is the half opening angle. These parameters are shown in figure 2.12. The expressions for $t(u,v)$ and $\theta(u,v)$ can be found by analyzing the cone's base shape which is a circular sector:

$$t(u,v) = H - \sqrt{u^2 + (H-v)^2} \tag{2.46}$$

$$\theta(u,v) = \frac{1}{\sin\gamma}\arctan\left(\frac{u}{H-v}\right) \tag{2.47}$$

The tangents of the local angles are defined in equations 2.24 and 2.25, and can be calculated from equation 2.45:

$$\tan\alpha_S = -\frac{(H-v)\left[H\sin^2\gamma - t(u,v)\right]}{[H - t(u,v)]\,H\cos\gamma\sin\gamma} \tag{2.48}$$

$$\tan\beta_S = -\frac{u\left[H\sin^2\gamma - t(u,v)\right]}{[H - t(u,v)]\,H\cos\gamma\sin\gamma} \tag{2.49}$$

The reconstructed conical surface in this example lies on a cone with an opening angle of $\gamma = 20°$ and a distance from the center of the surface to the tip of the

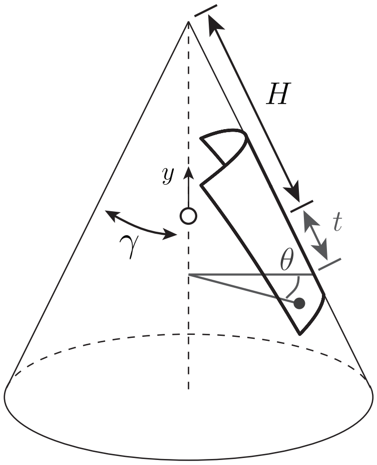


Figure 2.12: Definition of the conical surface parameters. The origin of the 3D coordinate system is the circle aligned with the 'center' of the surface located at a distance $H$ from the tip. A point on the surface (grey circle) is parametrized by the distance $t$ and the angle $\theta$.

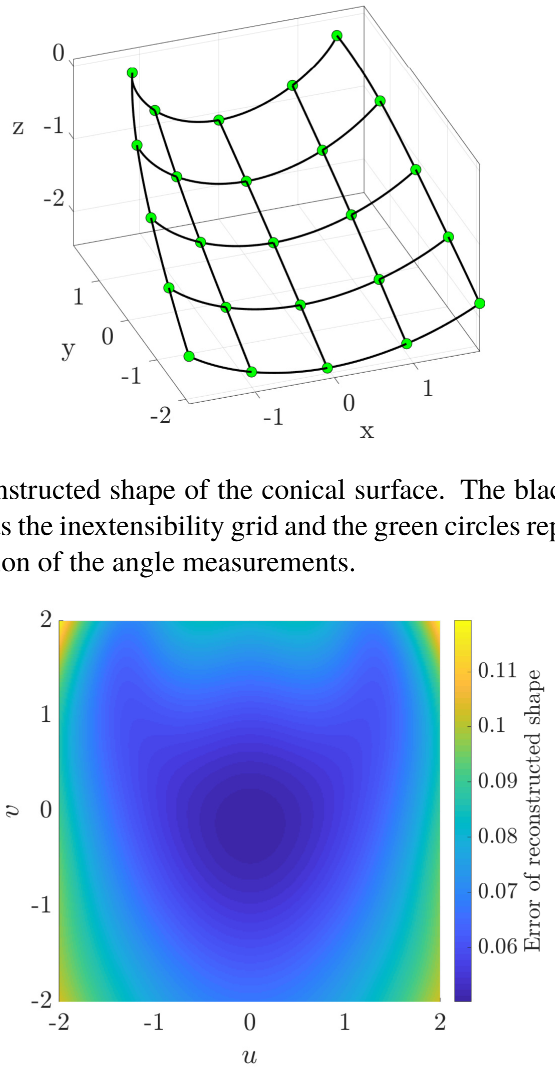
cone of $H = 5$. Figure 2.13a shows the reconstructed shape calculated using Lagrange polynomials and identical grids as in the previous example. As before, to avoid singularities (rigid-body rotations) the central point of curvilinear coordinates $(u, v) = (0, 0)$ has its x and y coordinates fixed to 0 while the point with curvilinear coordinates $(u, v) = (0, 1)$ has its x-coordinate fixed to 0. The reference configuration of the surface is a flat plane perpendicular to the z-axis:

$$\boldsymbol{r}^{\mathbf{0}}(u, v) = \begin{bmatrix} u \\ v \\ -H \sin \gamma \end{bmatrix} \tag{2.50}$$

The z-coordinate is equal to the distance of the center of the surface to the axis of the cone. This reference configuration prevents the spherical singularity to affect the convergence of the algorithm.

Figure 2.13b shows the error of the reconstruction compared to the analytical conical shape $\tilde{\boldsymbol{r}}$. It is minimal at the center of the surface where the two constrained points are located. Contrary to the cylinder case (Figure 2.10b), the error varies along $u$ and $v$ as the surface has to deform in both directions.

Figure 2.14 shows the distribution of the strain components. The strain errors increase as the curvature of the surface increases. This is due to the fact that Lagrange polynomials cannot perfectly reconstruct an ellipse. A Taylor expansion



(a) Reconstructed shape of the conical surface. The black grid represents the inextensibility grid and the green circles represent the location of the angle measurements.

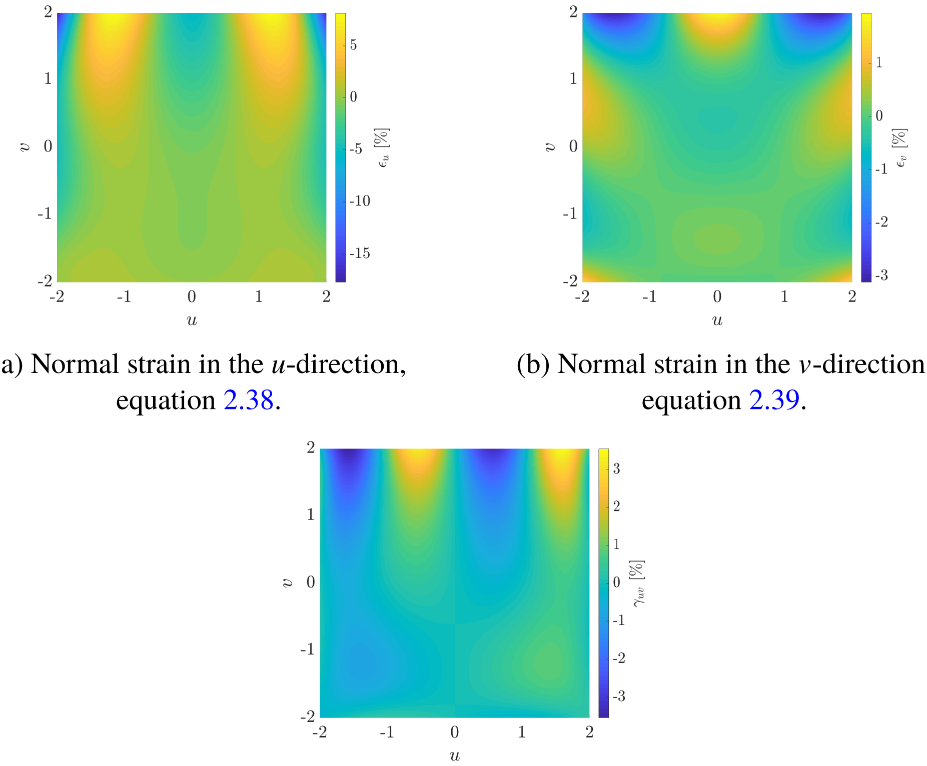
(b) Error between reconstructed conical surface shown in figure 2.13a and cone defined in equation 2.45.

Figure 2.13: Reconstruction of a conical surface with half-opening angle $\gamma = 20°$ and height $H = 5$ aligned with the y-axis using Lagrange polynomials on uniform grids $[\boldsymbol{u_k}] = [\boldsymbol{v_l}] = [\boldsymbol{\hat{u}_i}] = \left[\boldsymbol{\hat{v}_j}\right] = (-2, -1, 0, 1, 2)$.

shows that this error is on the order of $\kappa^{n-1}$ where $\kappa$ is the local curvature and $n$ is the degree of the Lagrange polynomial.

### Convergence of the Solution

In order to lower the errors, we studied the effect of refining the mesh of control points, but continuing to use uniform grids. Since the number of control points is increased, the number of angle measurements needs to increase to avoid any



(a) Normal strain in the $u$-direction, equation 2.38.

(b) Normal strain in the $v$-direction, equation 2.39.

(c) Shear strain, equation 2.40.

Figure 2.14: Strains of the reconstructed conical surface of half-opening angle 20° and height $H = 6$ aligned with the y-axis using Lagrange polynomials on uniform grids $[\boldsymbol{u_k}] = [\boldsymbol{v_l}] = [\boldsymbol{\hat{u}_i}] = \left[\boldsymbol{\hat{v}_j}\right] = (-2, -1, 0, 1, 2)$.

rank deficiency. The same grid is used for the interpolation points of the Lagrange polynomials, inextensibility constraints and angle measurements.

The convergence results for the cylinder of radius 2 are shown in figure 2.15. The mesh was only refined in the $u$-direction as the result does not depend on $v$. It was varied from a $3 \times 5$ grid to an $11 \times 5$. The x-axis shows the size of the grid in the $u$-direction. The y-axis shows the RMS error between each reconstructed shape and the analytical cylinder in log scale. We see that, as the grid gets more refined, the accuracy of the algorithm is improved, as expected. The relation is inversely exponential.

The convergence results for the cone are shown in figure 2.16. The mesh is refined in both $u$ and $v$ directions using grids ranging from $3 \times 3$ to $11 \times 11$. The accuracy also improves as the mesh gets denser.

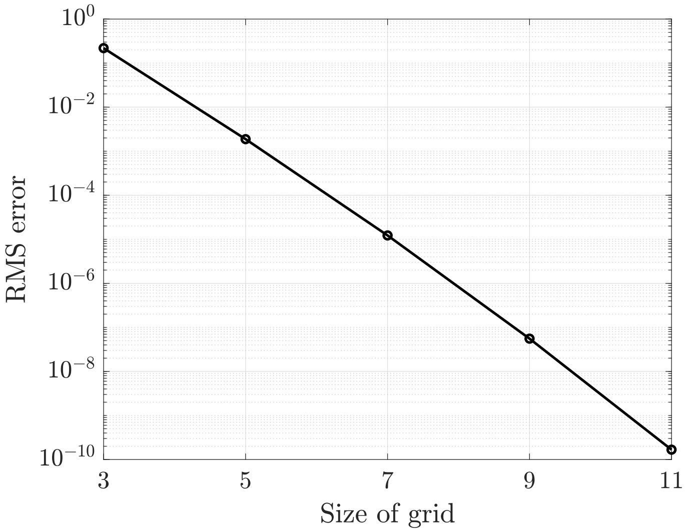


Figure 2.15: Variation of the RMS error between the reconstructed shape and the perfect cylinder by varying the size of the grid in the *u*-direction from 3 to 11 control points.

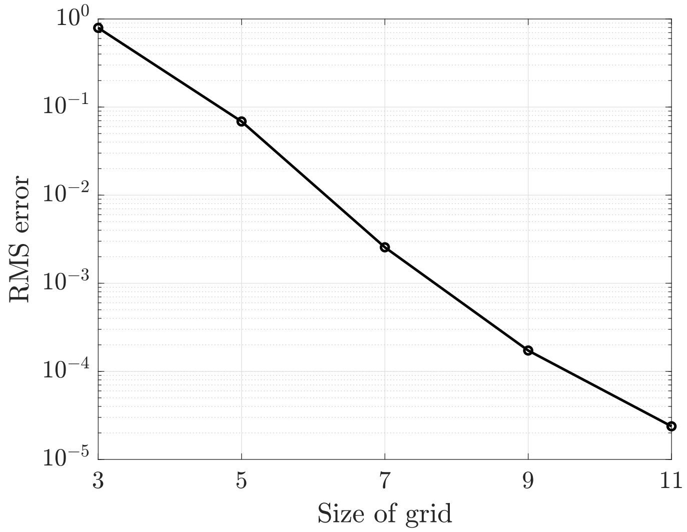


Figure 2.16: Variation of the RMS error between the reconstructed shape and the analytical cone by varying the size of the grid from $3 \times 3$ to $11 \times 11$ control points.

### Variation of Algorithm Parameters

For a fixed number of control points, the solution depends on three sets of parameters: the number and position of the angle sensors, the coordinates of the inextensibility grid, and the weight of each equation of system 2.31. By changing these parameters, the algorithm converges to different solutions that will have different RMS errors. We varied all parameters for a fixed $5 \times 5$ grid of control points, and studied their impact on the RMS error of the reconstructed cylinder and cone.

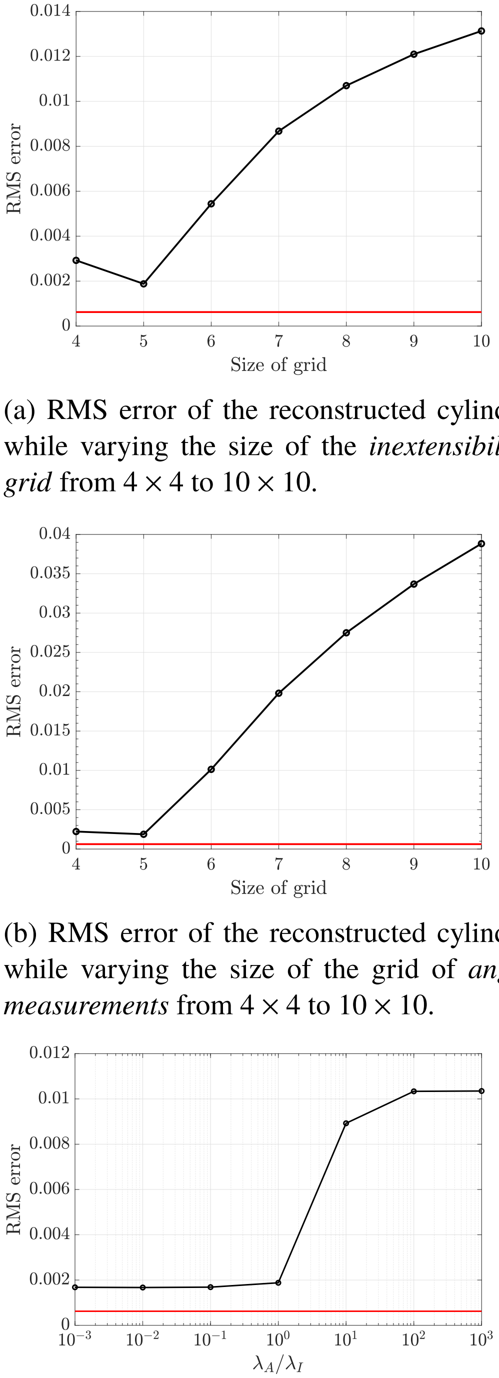
The results are shown in figures 2.17 and 2.18, where the red lines show the minimum RMS error possible. It corresponds to the optimal position of the $5 \times 5$ control points that minimizes the RMS error between the reconstructed shape and the exact cylinder or cone. This optimal result was obtained from an unconstrained minimization algorithm (*fminunc* in MATLAB).

The size of the inextensibility grid was varied from $4 \times 4$ to $10 \times 10$. The minimum grid size was dictated by the minimum number of equations needed in order to have more constraints than unknowns. The grid of control points and angle measurements were maintained at $5 \times 5$ for each calculation. Figure 2.17a shows the RMS error of the reconstructed cylinder of radius 2 for different grid sizes. Figure 2.18a shows the RMS error of the reconstructed cone. Note that both curves reach a minimum error and then the error increases.

Varying the number of angle measurements was performed by spreading the sensors on a uniform grid whose size ranges from $4 \times 4$ to $10 \times 10$ while using fixed $5 \times 5$ control points and inextensibility grids. The minimum grid size is again dictated by the minimum number of equations needed for a well-posed problem. Figure 2.17b shows the evolution of the RMS error of the reconstructed cylinder while figure 2.18b shows the evolution of the RMS error of the reconstructed cone. Note that the same trend noted above is seen in the error variation. The increase of RMS error after the minimum is however not monotonic for the conical shape. The error starts decreasing again for denser grids.

As stated in section 2.5, the weight of each equation of the system is important. We investigate the difference of weight between the first 3 equations, which relate to the inextensibility of the surface and the last two, which impose constraints on the measured angles.

It is important to note that the equations are not adimensional and different size of grids can affect the scaling of the inextensibility constraints. The first two equations of system 2.31 can easily be made adimensional by dividing by the length of the edges of the inextensibility grid in the reference configuration. Note that in our examples, the length of the edges is on the order of 1 and this division has almost no impact. The 3$^{\text{rd}}$ inextensibility equation could be made adimensional by dividing by the norm of the tangent vectors. The angle measurement constraints are properly scaled and do not depend on the size of the grids.

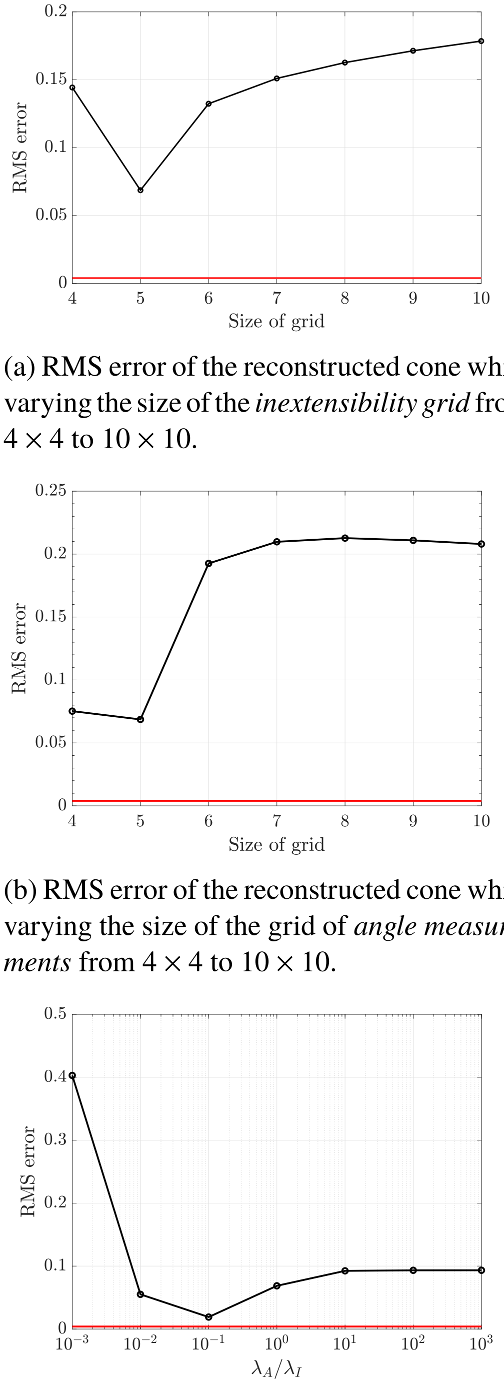


(a) RMS error of the reconstructed cylinder while varying the size of the *inextensibility grid* from 4 × 4 to 10 × 10.

(b) RMS error of the reconstructed cylinder while varying the size of the grid of *angle measurements* from 4 × 4 to 10 × 10.

(c) RMS error of the reconstructed cylinder while varying the *weight of the angle constraints* to the inextensibility constraints.

Figure 2.17: RMS error of the reconstructed cylinder while varying different parameters of the algorithm using a 5 × 5 grid of control points and Lagrange polynomials. The red line represents the minimum RMS error achievable using such basis functions ($6.2 \times 10^{-4}$).



(a) RMS error of the reconstructed cone while varying the size of the *inextensibility grid* from $4 \times 4$ to $10 \times 10$.

(b) RMS error of the reconstructed cone while varying the size of the grid of *angle measurements* from $4 \times 4$ to $10 \times 10$.

(c) RMS error of the reconstructed cone while varying the *weight of the angle constraints* to the inextensibility constraints.

Figure 2.18: RMS error of the reconstructed cone while varying different parameters of the algorithm using a $5 \times 5$ grid of control points and Lagrange polynomials. The red line represents the minimum RMS error achievable using such basis functions $(4.0 \times 10^{-3})$.

Figure 2.17c shows the influence of scaling the constraints when reconstructing a cylinder of radius 2. The tolerance limit $\epsilon$ defined in equation 2.35 was set to $10^{-6}$ for all simulations. The inextensibility conditions were multiplied by a scaling factor $\lambda_I = 1$ and the angle constraints by $\lambda_A$ which varied from $10^{-3}$ to $10^3$. Results for the cone are shown in figure 2.18c. The curves follow similar trends as the previous graphs and we notice that changing weights has only a small effects on the RMS error, especially for the cylinder.

The variations of the RMS error for these 3 different studies can be explained by analyzing the first two equations of system 2.31. The extension of the edges of the inextensibility grid is the main source of error variations in this system. Figure 2.19 shows how the norm of the error of the length equations varies between the algorithm and optimal solutions while reconstructing the cylinder:

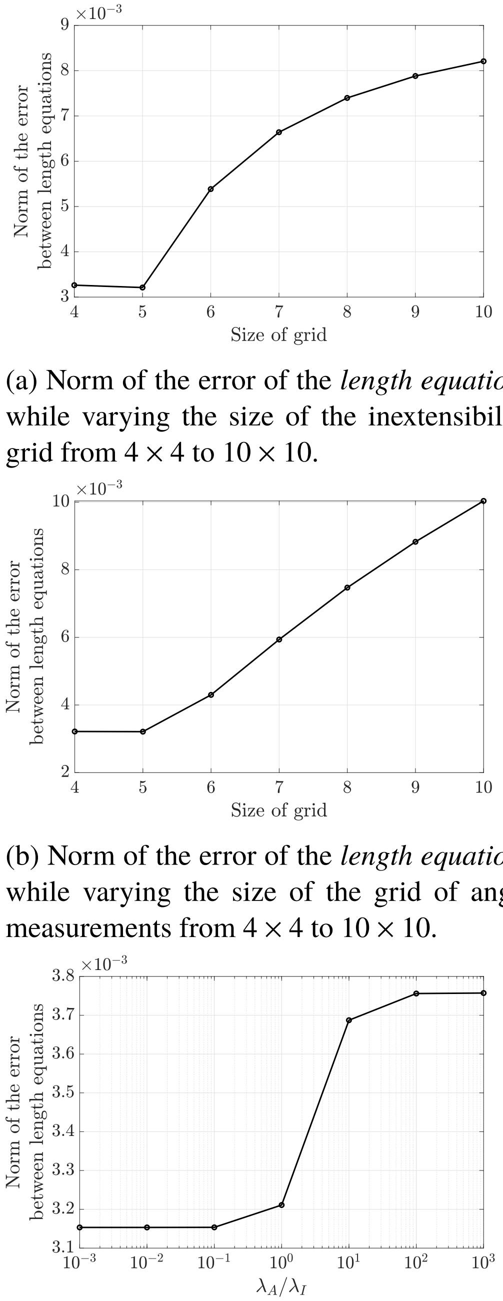
$$\text{Norm of the error} = \left\| \boldsymbol{f}_{u,v}(\bar{\boldsymbol{q}}^{**}) - \boldsymbol{f}_{u,v}(\bar{\boldsymbol{q}}^{*}) \right\| \tag{2.51}$$

where $\boldsymbol{f}_{u,v}$ represents the vector of the first two equations of system 2.31, $\bar{\boldsymbol{q}}^{**}$ is the optimal solution that minimizes the RMS error (red line in figures 2.17 and 2.18), and $\bar{\boldsymbol{q}}^{*}$ is the solution of the algorithm. We see that the curve is very similar to figure 2.17.

Changing the size of the inextensibility grid has a direct impact on this error. The optimal grid can be calculated on the optimal solution by making sure each edge retains its length from the reference configuration. As a result, the optimal solution will satisfy the defined inextensibility conditions and will be very close to the solution of the algorithm.

Increasing the number of angle measurements or the weight associated with their constraints forces the algorithm to violate the inextensibility conditions in order to satisfy an increasing number of angle constraints (or a more weighted set of equations). Because Lagrange polynomials cannot precisely fit circular segments, even the optimal solution is not perfectly inextensible. The solution of the algorithm is close to the optimal one if it violates these constraints by the same magnitude. There is an optimal algorithm where the amplitude of the error of the inextensibility conditions coincides with the error of the optimal solution.

Finally, an optimization program could be developed to optimize the parameters of the algorithm to minimize the RMS error for specific, desired shapes. Results presented in this chapter show that matching the inextensibility grid with the control points and the angle sensors leads to a near optimal scheme.



(a) Norm of the error of the *length equations* while varying the size of the inextensibility grid from 4 × 4 to 10 × 10.

(b) Norm of the error of the *length equations* while varying the size of the grid of angle measurements from 4 × 4 to 10 × 10.

(c) Norm of the error of the *length equations* while varying the weight of the angle constraints to the inextensibility constraints.

Figure 2.19: Norm of the error of the length equations (first 2 equations of system 2.31) on the algorithm solution and optimal solution.

## 2.7 Conclusion and Discussion

In this chapter, we have presented a mathematical model to reconstruct the shape of a 3D surface based on a template and the angle measurements from embedded sensors. The template is a known configuration of the surface and it is assumed that it deforms inextensibly to its current configuration.

One of the advantages of this model is to be able to use a relatively small number of data to reconstruct the surface. The number of equations to solve is therefore small which can reduce computational times making this algorithm solvable in real time.

Aerospace is one of the application areas for this method. Sun sensors are already widely used in this industry for attitude control, and together with this algorithm, could be used to measure the shape of a deployable or reconfigurable structure in space (such as solar sails or large antenna arrays).

The main limitation of the algorithm resides in a numerical singularity associated with the distance of the surface to the light source. This is why practical applications usually require the light to be at an infinite distance (like the Sun) or the use of sensors that measure the angles from a fixed direction. Wearable technologies used for augmented reality, medical purposes or robotics are other applications of this method using such sensors (usually IMUs).

Only developable surfaces have been considered in this study. More work should be done to understand the performance of the presented algorithm on more complex shapes such as doubly curved surfaces and surfaces with localized deformations like kinks or buckles.

*Chapter 3*

# INEXTENSIBLE SURFACE RECONSTRUCTION FROM DISTRIBUTED ANGLE MEASUREMENTS: EXPERIMENT USING LIGHT SENSORS

## 3.1 Introduction

Measuring the shape of a surface often requires an imaging system placed in front of the structure. Sets of cameras or laser ranging devices have been used to generate a point cloud of the surface which can be reconstructed into a surface [8, 46]. Simpler hardware such as a single camera can also be used in what is called monocular surface reconstruction [15, 59, 73].

Such techniques require a minimum depth of view in front of the surface to measure. In some cases, however, this space might not be available. The surface can be too large compared to the field of view of the instruments or there might be no space available in front of the surface. Shallow angle cameras have been used to measure the displacement of trackers on a surface with millimeter accuracy [17, 51]. Such methods are, however, susceptible to shading (physical blocking of the view of the camera due to the deformation), light conditions and use relatively complex image processing softwares which can limit their application.

This chapter presents a different approach with embedded vision sensors inside the surface to be reconstructed. They measure the local angles at specific locations in the structure, whose shape can then be recovered by spatial integration. Similar to monocular shape reconstruction algorithms, ambiguities appear as shown in chapter 2. Angle measurements alone are not sufficient to reconstruct the shape of the surface as depth effects cannot be solved for: the distance between sensors needs to be known to perform a spatial integration. To remove these singularities, the deformation is assumed inextensible from a known reference configuration (see figure 3.1). This approach was used in similar research [15, 38, 40, 59, 73, 80].

Different embedded sensors have been used to reconstruct surfaces. Inertial sensors (a combination of accelerometers and magnetometers) for instance, measure the angles from the gravity directions and Earth's magnetic field. Both 3D curves and surfaces have been reconstructed by distributing such sensors on the support structure [38–40, 80, 81].

Inertial Measurement Units (IMUs), or light sensors have been used mostly as standalone components to measure the orientation of a spacecraft [61] or a solar panel [23, 95]. Recently, sun sensors have been investigated to reconstruct large surfaces in space [10, 84, 85]. Compared to IMUs, they have the advantage of being insensitive to accelerations or magnetic field disturbances, hence being able to measure shapes in space.

Many light sensor designs exist. A review of different designs is available in [71]. Single photodiodes can estimate the angle from a known light as the measured intensity varies with the cosine of the angle. They can only measure the angle of the light source from their normal direction and more complex architectures as in [5] require a 2-angle measurement. Most solutions combine an array of photodiodes or pixels with an aperture on top [19, 23, 25, 60, 61, 91]. The photo-sensitive element ranges from a simple quad-photodiode to a CMOS or CCD image sensor (in miniature camera systems) [11, 50, 57, 94]. Their precision can vary from a few degrees to under an arc minute. Usually limited by their field-of-view, techniques exist to expand the field-of-view by having multiple aperture holes [23, 91]. Such sensors are precise but can be relatively large and power hungry. While many designs have been engineered, most of them tend to be bulky and expensive. The presented application requires many sensors to be placed on a flexible structure. A small, simple, lightweight design that still achieves good performance and is better suited for this type of application is introduced in this chapter.

Section 3.2 summarizes the algorithm that is used to reconstruct the shape of a surface assuming inextensibility of the deformation from an initially flat surface, and the angle measurements from the light sensors. It has been previously introduced in chapter 2.

The light sensor designed for the experiment is introduced in section 3.3. It consists of a quad-photodiode placed under a square aperture, effectively recreating a simple pinhole camera. The theoretical behavior of the sensor is detailed to understand the correlation between the measured intensities of the photodiodes and the angles and geometry of the sensor. From these equations, the optimal design that maximizes the sensitivity of the sensor is presented. The electronic and software architecture that reduces and filters the noise of the sensor is detailed.

Section 3.4 describes the experimental setup. The surface to be reconstructed is a 1.3 m × 0.25 m sheet of aluminum integrated with 14 light sensors. Its boundary conditions, the communication architecture between sensors and the illumination system are detailed. A second measurement setup is used to measure the shape of the surface with better accuracy in order to assess the performance of the light sensor system.

Before performing the experiment, each sensor is calibrated as described in section 3.5. The calibration is used to estimate the error between the actual response of the sensor and its theoretical behavior described in section 3.3. The noise of the sensor is also investigated. A calibration was eventually performed on all sensors to increase their precision. The repeatability and accuracy of the calibrations are assessed to understand the performance limits of the sensor.

Finally the results of the experiment are presented in section 3.6. The surface is deformed in bending and torsion. To understand the overall accuracy of the system, the RMS error is calculated by comparing the reconstructed shape to an accurate measurement provided by a secondary, optical measurement system. The RMS error follows a Generalized Extreme Value distribution and, on average, is on the order of a millimeter and can be as low as 0.5 mm. We show the evolution of the RMS error with the complexity of the shape and identify different factors that limit the accuracy of the system.

## 3.2 Reconstruction Algorithm

The algorithm used to reconstruct the shape of a surface from angle measurements was first studied in chapter 2. It consists of a set of equations constraining the surface to match the measured angles at the sensor location while forcing its inextensibility from a reference configuration. These equations are summarized in the first sub-section.

The second sub-section details the modifications required to adapt the algorithm to real-world applications and specifically to the experiment introduced in this chapter. The thickness of the structure, the offset of the sensors from the surface and the size of the sensors are all taken into account to reconstruct the experimental structure.

### Reconstruction of a Theoretical Surface

The following algorithm reconstructs a 3D surface whose shape is discretized on a finite set of basis functions. It was first introduced in chapter 2. A point on the surface is described by its two curvilinear coordinates $u, v$ (see figure 3.1). The

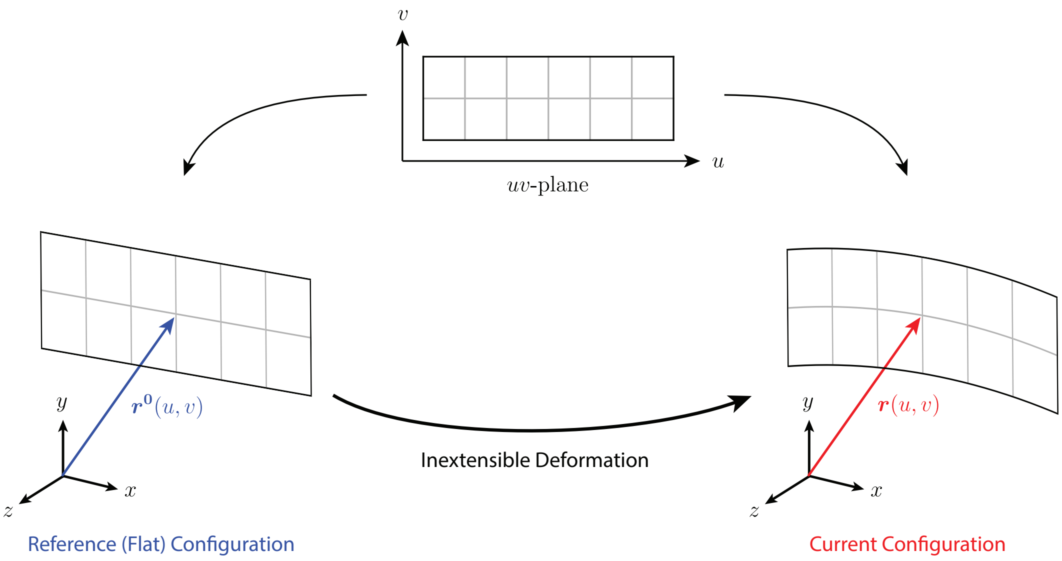


Figure 3.1: Definition of the problem. The surface is parametrized by two coordinates $(u, v)$. The 3D surface is a mapping of the 2D coordinates to 3D. The shape of the reference configuration is known while the current configuration needs to be reconstructed.

current 3D position of that point is written as $\boldsymbol{r}(u, v)$ such that:

$$\begin{aligned} \boldsymbol{r} : X \subset \mathbb{R}^2 &\longrightarrow \mathbb{R}^3 \\ (u, v) \quad &\longmapsto \boldsymbol{r}(u, v) = \sum_{k=1}^{N} \boldsymbol{q_k} \phi_k(u, v) \end{aligned} \tag{3.1}$$

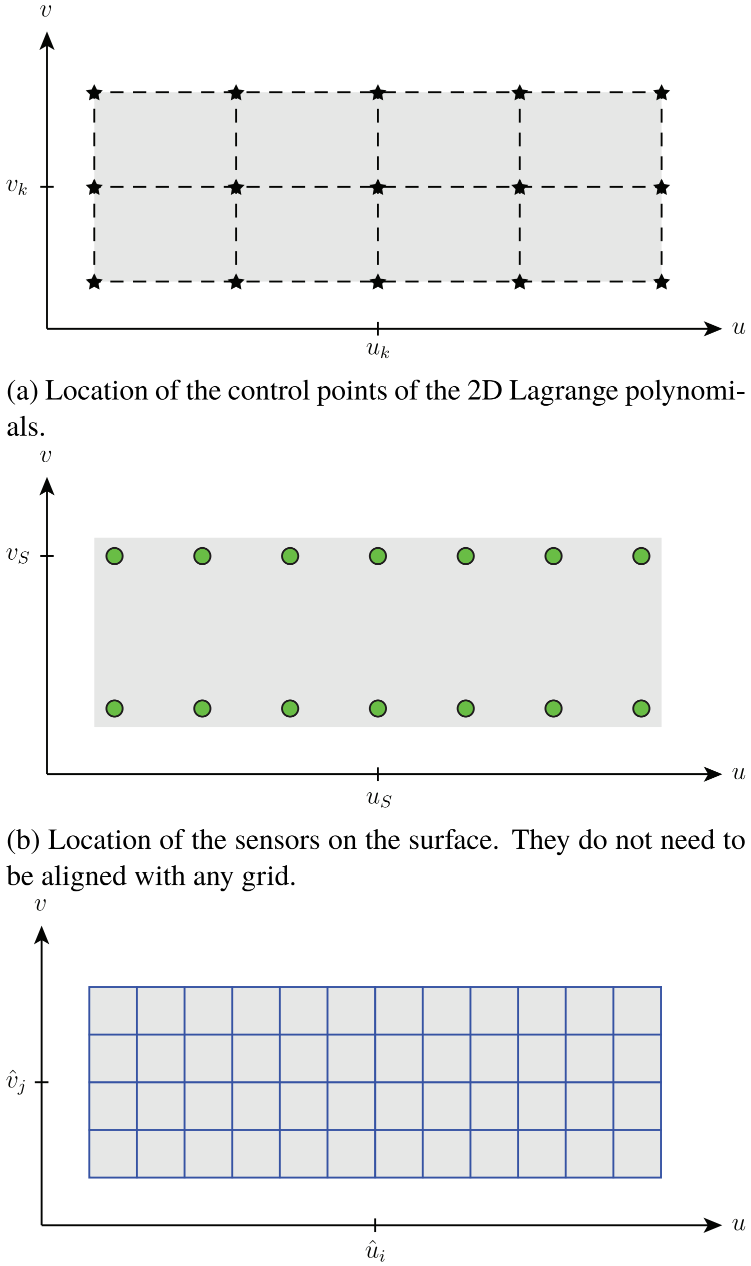
where $\phi_k : X \longrightarrow \mathbb{R}$ are basis functions, $\boldsymbol{q_k}$ are unknown 3D points called control points, and $N$ is the dimension of the function space.

The basis functions selected for the reconstructions shown in this chapter are 2D Lagrange functions weighted by the control points $\boldsymbol{q_k}$ arranged in a 2D grid in the $(u, v)$-space (see grid in figure 3.2a).

The position of the control points is calculated from a set of constraints derived from the measurement of the sensors and the inextensibility of the deformation.

Sensors measure the angles between the local normal of the structure and the light rays coming from a point source located at the origin of 3D coordinate system. If the sensor is located on the surface at a location $(u_S, v_S)$ (see figure 3.2b), the tangent of these angles can be calculated from:

$$\tan \alpha_S = \frac{\boldsymbol{r}(u_S, v_S) \cdot \frac{\partial \boldsymbol{r}}{\partial v}(u_S, v_S)}{\boldsymbol{r}(u_S, v_S) \cdot \boldsymbol{n}(u_S, v_S)} \frac{\|\boldsymbol{n}(u_S, v_S)\|}{\left\|\frac{\partial \boldsymbol{r}}{\partial v}(u_S, v_S)\right\|} \tag{3.2}$$



(a) Location of the control points of the 2D Lagrange polynomials.

(b) Location of the sensors on the surface. They do not need to be aligned with any grid.

(c) Inextensibility grid. Each edge retains its length upon transformation while the angle around each vertex is constant.

Figure 3.2: Parameters of the algorithm defined in the *uv*-space.

$$\tan \beta_S = -\frac{\boldsymbol{r}(u_S, v_S) \cdot \frac{\partial \boldsymbol{r}}{\partial u}(u_S, v_S)}{\boldsymbol{r}(u_S, v_S) \cdot \boldsymbol{n}(u_S, v_S)} \frac{\|\boldsymbol{n}(u_S, v_S)\|}{\left\|\frac{\partial \boldsymbol{r}}{\partial u}(u_S, v_S)\right\|} \tag{3.3}$$

where $\alpha$ is the angle around the $u$-direction, $\beta$ around the $v$-direction, $\boldsymbol{n}(u_S, v_S) = \frac{\partial \boldsymbol{r}}{\partial u}(u_S, v_S) \times \frac{\partial \boldsymbol{r}}{\partial v}(u_S, v_S)$ is the normal of the surface at the sensor location, and $\boldsymbol{r}(u_S, v_S)$ coincides with the direction of the light.

Inextensibility conditions are derived from a discretization of the metric tensor. This tensor is used in differential geometry to measure the length of segments on curved surfaces [1, 47]. By imposing the condition that the metric tensor does not change before and after the transformation, the length of any segment drawn on the surface stays the same. Such an implementation requires the knowledge of the shape of the surface before the transformation. This configuration is called reference configuration and it is assumed to be flat, as shown in figure 3.1.

A discretization of the metric tensor is performed by defining a regular grid on the surface (see figure 3.2c). The coordinates of the grid are $\hat{u}_i$ and $\hat{v}_j$. Each edge of the grid is constrained to have the same length in the current and reference configurations. The angles at each vertex of the grid are also conserved. This is done by conserving the dot product of the tangent vectors of the surface. As a result, the following constraints are written:

$$\int_{\hat{u}_i}^{\hat{u}_{i+1}} \left\| \frac{\partial \boldsymbol{r}}{\partial u}(u, \hat{v}_j) \right\| du = \int_{\hat{u}_i}^{\hat{u}_{i+1}} \left\| \frac{\partial \boldsymbol{r}^{\boldsymbol{0}}}{\partial u}(u, \hat{v}_j) \right\| du \tag{3.4}$$

$$\int_{\hat{v}_j}^{\hat{v}_{j+1}} \left\| \frac{\partial \boldsymbol{r}}{\partial v}(\hat{u}_i, v) \right\| dv = \int_{\hat{v}_j}^{\hat{v}_{j+1}} \left\| \frac{\partial \boldsymbol{r}^{\boldsymbol{0}}}{\partial v}(\hat{u}_i, v) \right\| dv \tag{3.5}$$

$$\frac{\partial \boldsymbol{r}}{\partial u}^T (\hat{u}_i, \hat{v}_j) \frac{\partial \boldsymbol{r}}{\partial v}(\hat{u}_i, \hat{v}_j) = \frac{\partial \boldsymbol{r}^{\boldsymbol{0}}}{\partial u}^T (\hat{u}_i, \hat{v}_j) \frac{\partial \boldsymbol{r}^{\boldsymbol{0}}}{\partial v}(\hat{u}_i, \hat{v}_j) \tag{3.6}$$

where $\boldsymbol{r}^{\boldsymbol{0}}$ is the reference, known configuration of the surface.

By gathering the measurement equations 3.2 and 3.3 written at each sensor location and the inextensibility equations 3.4, 3.5, and 3.6 for each segment or node of the inextensibility grid, a system of equations with $3N$ unknowns can be written (the three coordinates of the control points). It is usually overconstrained and can be solved using a Levenberg-Marquardt algorithm.

### Practical Implementations

Modifications to the previous equations need to be implemented to account for the thickness of the surface, the offset of the sensors from the surface and their size. Section 3.4 details the design of the experiment, where these dimensions can be better understood and appreciated.

The surface previously defined by $\boldsymbol{r}(u, v)$, where both the control points and the inextensibility grid are located, now corresponds to the mid-plane of the structure.

This ensures that the mid-plane strains in the surface remain zero. As a result, the inextensibility constraints remain valid.

If the top of the structure is of interest rather than its mid-plane, an offset of half the thickness along the local normal can be applied (assuming the Kirchhoff hypothesis for thin plates [21]):

$$\boldsymbol{s}(u,v) = \boldsymbol{r}(u,v) + \frac{t}{2}\frac{\boldsymbol{n}(u,v)}{\|\boldsymbol{n}(u,v)\|} \tag{3.7}$$

where $\boldsymbol{s}(u,v)$ is the top surface, $t$ is the thickness of the structure, and $\boldsymbol{n}(u,v)$ is the local normal.

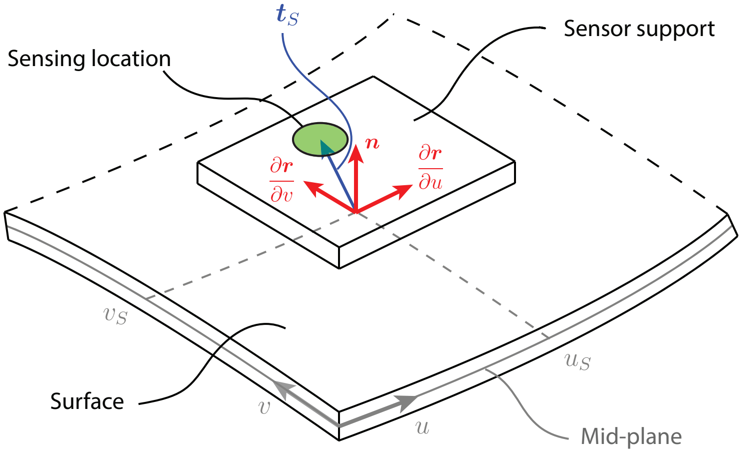


Figure 3.3: Offset of a sensor from the mid-plane of the structure. The fixed offset $\boldsymbol{t}_S$ is defined in the local reference system (in red).

The only equations that need to be modified are the ones related to the angle measurements (equations 3.2 and 3.3). The sensor is not located on the mid-plane, but is offset by a fixed distance. Figure 3.3 defines the offset of a sensor from the mid-plane. Furthermore, the sensors are assumed to be rigid which locally prevents the surface from curving. This effect cannot be captured by the Lagrange basis functions. As a result, the local tangent and normal vectors needed in the equations are calculated in the middle of the rigid support of the sensors. This approximates the actual direction of the sensor. The sensing element is offset in this local reference system by a fixed vector $\boldsymbol{t}_S$ (see figure 3.3). The angle equations become:

$$\tan\alpha_S = \frac{\boldsymbol{d}_S \cdot \frac{\partial \boldsymbol{r}}{\partial v}(u_S, v_S)}{\boldsymbol{d}_S \cdot \boldsymbol{n}(u_S, v_S)} \frac{\|\boldsymbol{n}(u_S, v_S)\|}{\|\frac{\partial \boldsymbol{r}}{\partial v}(u_S, v_S)\|} \tag{3.8}$$

$$\tan\beta_S = -\frac{\boldsymbol{d}_S \cdot \frac{\partial \boldsymbol{r}}{\partial u}(u_S, v_S)}{\boldsymbol{d}_S \cdot \boldsymbol{n}(u_S, v_S)} \frac{\|\boldsymbol{n}(u_S, v_S)\|}{\|\frac{\partial \boldsymbol{r}}{\partial u}(u_S, v_S)\|} \tag{3.9}$$

with $\boldsymbol{d}_S$ the vector between the center of the 3D reference system (where the light source is located) and the center of the light sensor, offset by $\boldsymbol{t}_S$ in the local reference system. This vector can be written as:

$$\boldsymbol{d}_S = \boldsymbol{r}(u_S, v_S) + \mathbf{R_S}\boldsymbol{t}_S \tag{3.10}$$

where $\mathbf{R_S}$ is the local rotation matrix at the sensor location which is defined as the concatenation of the tangent and normal unit vectors:

$$\mathbf{R_S} = \begin{bmatrix} \frac{\frac{\partial \boldsymbol{r}}{\partial u}(u_S,v_S)}{\left\|\frac{\partial \boldsymbol{r}}{\partial u}(u_S,v_S)\right\|} & \frac{\frac{\partial \boldsymbol{r}}{\partial v}(u_S,v_S)}{\left\|\frac{\partial \boldsymbol{r}}{\partial v}(u_S,v_S)\right\|} & \frac{\boldsymbol{n}(u_S,v_S)}{\|\boldsymbol{n}(u_S,v_S)\|} \end{bmatrix} \tag{3.11}$$

## 3.3 Light Sensor Design

### Basic Principles

A schematic of the sensor used for the experiment is shown in figure 3.4. It consists of an aperture placed on top of a quad-photodiode. The square aperture is placed at a height $h$ above the photosensitive area. The aperture has a thickness $t$ and a size $d$. It is located above the center of the array. Each photodiode is square and has a length $L$. It is separated by a gap of size $e$ from its neighbor. The sensor is rotated from the incoming light by the angles $\alpha$ and $\beta$ (defined in figure 3.4). It is assumed that there is no intermediate layer (glass or clear plastic) between the aperture and the photodiodes that would refract the light at another angle.

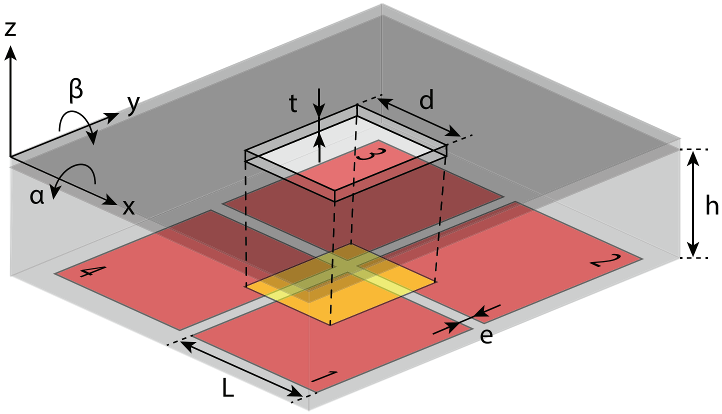


Figure 3.4: Drawing of a light sensor made from a quad-photodiode (red) and a square aperture of length $d$. The thickness of the aperture is $t$ while its distance from the photosensitive plane is $h$. Each square photodiode has a length $L$ and is separated by a distance $e$ from its neighbours.

It is assumed that all the photodiodes have the same responsivity and that the gain applied on their output is also identical. Let $A$ be the responsivity of the photodiodes in amperes per watt (A/W), $W_0$ the incoming light intensity in watts (W) and $K$ the gain of the amplification circuit in arbitrary unit per amperes. Using the naming convention of figure 3.4, and assuming that the light is coming uniformly from the z-direction, the measured light intensity of each photodiode is:

$$\begin{split} I_1 = KAW_0 \cos\gamma \left( \frac{d-e}{2} + (h + t \ \mathrm{H}(-\alpha)) \tan\alpha \right) \times \\ \left( \frac{d-e}{2} + (h + t \ \mathrm{H}(-\beta)) \tan\beta \right) \end{split} \tag{3.12}$$

$$\begin{split} I_2 = KAW_0 \cos\gamma \left( \frac{d-e}{2} - (h + t \ \mathrm{H}(\alpha)) \tan\alpha \right) \times \\ \left( \frac{d-e}{2} + (h + t \ \mathrm{H}(-\beta)) \tan\beta \right) \end{split} \tag{3.13}$$

$$\begin{split} I_3 = KAW_0 \cos\gamma \left( \frac{d-e}{2} - (h + t \ \mathrm{H}(\alpha)) \tan\alpha \right) \times \\ \left( \frac{d-e}{2} - (h + t \ \mathrm{H}(\beta)) \tan\beta \right) \end{split} \tag{3.14}$$

$$\begin{split} I_4 = KAW_0 \cos\gamma \left( \frac{d-e}{2} + (h + t \ \mathrm{H}(-\alpha)) \tan\alpha \right) \times \\ \left( \frac{d-e}{2} - (h + t \ \mathrm{H}(\beta)) \tan\beta \right) \end{split} \tag{3.15}$$

where $\mathrm{H}(\cdot)$ is the Heaviside step function, and $\gamma$ is the angle of the light from the normal of the sensor which can be calculated from the angles $\alpha$ and $\beta$ using the Pythagorean theorem:

$$\tan(\gamma)^2 = \tan(\alpha)^2 + \tan(\beta)^2 \tag{3.16}$$

The coordinates of the centroid of the light spot can be calculated from these intensities. Let $R_\beta$ (resp. $R_\alpha$) be the normalized coordinates in the x-direction (resp. y-direction):

$$R_\alpha = \frac{(I_1 + I_4) - (I_2 + I_3)}{I_1 + I_2 + I_3 + I_4} = \frac{(2h + t) \tan\alpha}{d - e - t \ \mathrm{sgn}\alpha \tan\alpha} \tag{3.17}$$

$$R_\beta = \frac{(I_1 + I_2) - (I_3 + I_4)}{I_1 + I_2 + I_3 + I_4} = \frac{(2h + t) \tan\beta}{d - e - t \ \mathrm{sgn}\beta \tan\beta} \tag{3.18}$$

where $\mathrm{sgn}(\cdot)$ is the sign function. Each centroid is a function of only one of the angles. Additionally, the centroids do not depend on the light intensity or gains, hence a sensor can be used in various light environments without the need for calibrating it each time. The non-zero thickness of the aperture creates a non-linearity between the centroids and the tangent of the angles. The thinner the aperture mask, the more linear the response.

Because of the gap between photodiodes, it is important to have a square aperture in order to decouple the response of the sensor from the measured angles $\alpha$ and $\beta$, as seen in equations 3.17 and 3.18. This keeps the amount of light trapped in the gap constant when the angles of the sensor vary.

One can invert equations 3.17 and 3.18 to compute the tangent of the angles of the sensor in function of the measured centroids $R_\alpha$ and $R_\beta$:

$$\tan\alpha = \frac{(d-e)\,R_\alpha}{2h+t+t\ \mathrm{sgn}(R_\alpha)R_\alpha} \tag{3.19}$$

$$\tan\beta = \frac{(d-e)\,R_\beta}{2h+t+t\ \mathrm{sgn}(R_\beta)R_\beta} \tag{3.20}$$

where $R_\beta$ and $R_\alpha$ are calculated from the measured intensities (equations 3.17 and 3.18).

These equations are only valid while the light spot is within the boundaries of each photodiode. The field-of-view (FOV) of the sensor is defined as the maximum angle before the light spot hits the boundary of a photodiode. This can happen in two ways: 1) the spot can hit the inside boundary which is at a distance $e/2$ from the center or 2) the spot hits the outside boundary which is at a distance $e/2+L$ from the center as described in figure 3.5. As a result, the field of view can be written as the minimum of the two angles represented by each situation:

$$\tan\mathrm{FOV} = \min\left(\frac{d-e}{2\,(h+t)}, \frac{2L+e-d}{2h}\right) \tag{3.21}$$

The quad-photodiode selected for this experiment was the TE Electronics OPR5911, in which the space between the photodiodes and the aperture is filled with a transparent polymer. Therefore, the previous equations need to be modified to account for the refractive index of the polymer:

$$\tan\left[\arcsin\left(\frac{\sin\alpha}{n}\right)\right] = \frac{(d-e)\,R_\alpha}{2h+t+t\ \mathrm{sgn}(R_\alpha)R_\alpha} \tag{3.22}$$

$$\tan\left[\arcsin\left(\frac{\sin\beta}{n}\right)\right] = \frac{(d-e)\,R_\beta}{2h+t+t\ \mathrm{sgn}(R_\beta)R_\beta} \tag{3.23}$$

where $n$ is the refractive index of the polymer, estimated to be 1.56 (see section 3.5).

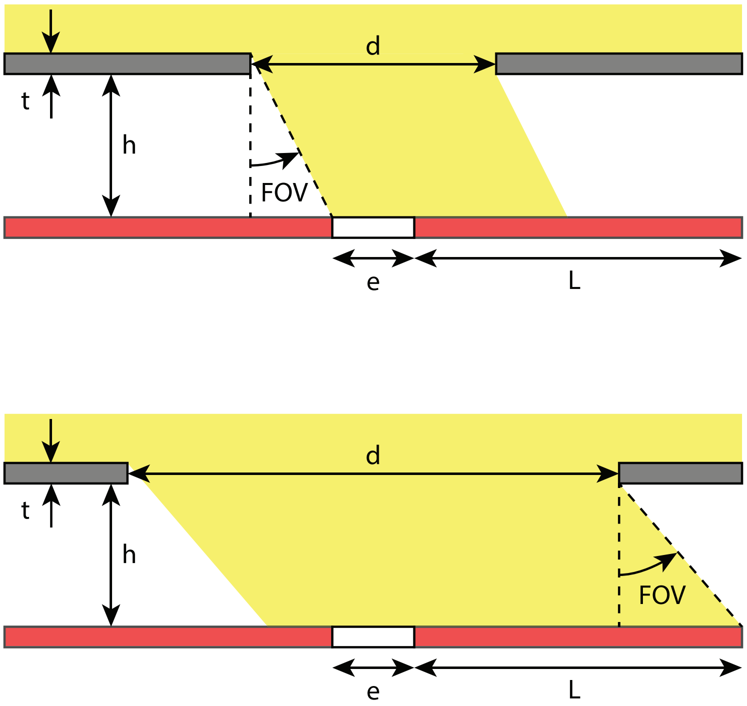


Figure 3.5: Two different cases defining the field-of-view of the sensor. Top: the spot hits the inside boundary of the photodiode. Bottom: the spot hits the outside boundary of the photodiode. The color red represents the photodiodes, grey represents the mask, and yellow represents the light.

### Optimal Design

The previous sub-section has identified five parameters that control the response of the light sensor to changes in angles: $d$, $e$, $h$, $t$, and $L$. Some of these parameters can be varied in order to change the field-of-view or the sensitivity of the output. The size of the photodiode array ($L$ and $e$) are fixed by the manufacturer. In our case, the TE Electronics OPR5911 was selected where $L = 1.27$ mm and $e = 0.25$ mm as measured under a microscope.

We study the influence of the 3 remaining parameters: the size $d$, height $h$ and thickness $t$ of the aperture hole. The sensitivity is defined as:

$$\mathrm{Sens}_\alpha(\tan\alpha) = \frac{dR_\alpha}{d\tan\alpha} = \frac{(2h+t)(d-e)}{(d-e-t\,\mathrm{sgn}\alpha\tan\alpha)^2} \tag{3.24}$$

$$\mathrm{Sens}_\beta(\tan\beta) = \frac{dR_\beta}{d\tan\beta} = \frac{(2h+t)(d-e)}{(d-e-t\,\mathrm{sgn}\beta\tan\beta)^2} \tag{3.25}$$

The goal is to maximize the sensitivity under a fixed field-of-view. Since the sensitivity depends on the angle, we decide to maximize the average sensitivity over the range $[\tan(-\mathrm{FOV}), \tan(+\mathrm{FOV})]$ which is equal to:

$$\begin{aligned} \overline{\mathrm{Sens}} &= \frac{R_{\alpha/\beta}(\mathrm{FOV}) - R_{\alpha/\beta}(-\mathrm{FOV})}{2\ \tan\mathrm{FOV}} \\ &= \frac{2h+t}{d-e-t\tan\mathrm{FOV}} \end{aligned} \tag{3.26}$$

Depending on the two cases defining the field-of-view (see figure 3.5), we end up with two different average sensitivities. It is possible to show that the maximum average sensitivity is equal to:

$$\overline{\mathrm{Sens}} = \frac{1}{\tan\mathrm{FOV}} \tag{3.27}$$

and corresponds to a field-of-view dictated by the light spot hitting the inside boundary of a photodiode.

We note that for a specific field-of-view and aperture height $h$, there is a maximum value for the thickness of the aperture (which corresponds to the two limits of the FOV being equal):

$$t_{\mathrm{max}} = \frac{L}{\tan\mathrm{FOV}} - 2h \tag{3.28}$$

As a result, in order to maximize the sensitivity of the sensor, the geometric parameters $h$, $t$, and $d$ need to satisfy the following equations:

$$\begin{cases} d & = e + 2(h+t)\tan\mathrm{FOV} \\ t & < \frac{L}{\tan\mathrm{FOV}} - 2h \end{cases} \tag{3.29}$$

and any set of parameters solving these equations corresponds to a sensor with maximum average sensitivity which is inversely equal to the tangent of the field-of-view (equation 3.27). Note that, because of the transparent polymer between the photodiodes and the aperture, the field-of-view in the previous equations need to be replaced by:

$$\mathrm{FOV}^* = \arcsin\left(\frac{\sin\mathrm{FOV}}{n}\right) \tag{3.30}$$

For our specific sensor, we chose $h = 0.43$ mm which corresponds to the height of the OPR5911 chip. As a result, the aperture can simply be put on the device which simplifies its assembly. The field-of-view of the sensor is set to 45° which is about 50% larger than the maximum angle seen during the experiment which leaves enough room for alignment errors. In order to be even more conservative, we neglected the refractive index of the polymer ($n = 1$). The aperture (figure 3.6) is manufactured out of a black anodized aluminum foil (Thorlabs BKF12) of 0.002" (50 $\mu$m) thickness which is less than the maximum thickness of 0.95 mm. The anodization prevents reflections inside the sensor. The size of the hole can be computed from equation 3.29 and is equal to 1.3 mm. The hole is cut by a Universal Laser System XLS10 laser cutter using 1.06 $\mu$m fiber laser.

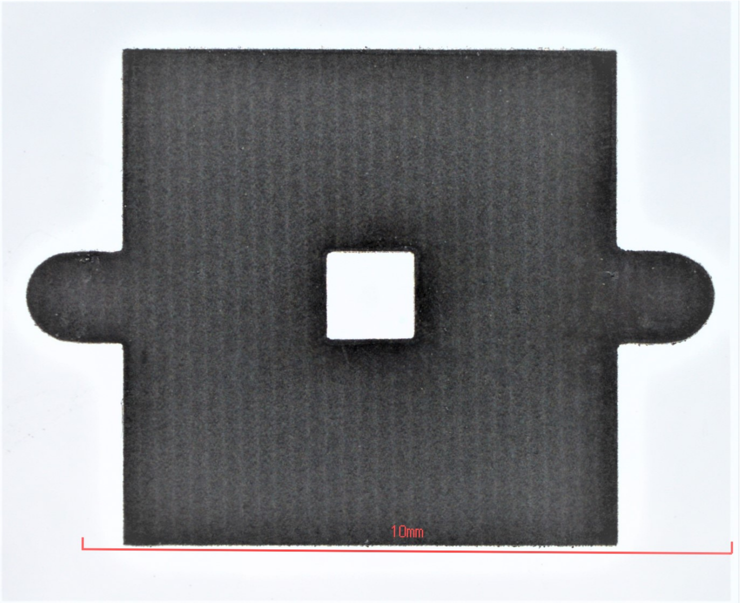


Figure 3.6: The aperture mask for the OPR5911 quad photodiode.

**Electronics Design (Contributed by Dr. Luke Chen and based on previous work published in [85])**

The light sensor electronic circuit consists of three major function blocks – the quad-photodiode, the amplifier circuit, and the microcontroller. The quad-photodiode is composed of four independent photodiodes arranged in the form of a 2-by-2 array. Each photodiode can convert light energy into electrical signal which can be measured in current and converted to a voltage using a common current-to-voltage converter implemented with an operational amplifier (figure 3.7a). The photodiode connected to the operational amplifier could have a reverse bias voltage (photoconductive mode) or zero bias voltage (photovoltaic mode). In general, the photoconductive mode has a faster and linear response, but could easily generate noisy signal because of the dark leakage current created by the reverse bias voltage, especially in low-light condition. The photovoltaic mode having slower response time, but less noise and temperature effect, meets the high-precision and low-frequency requirements of this research [62].

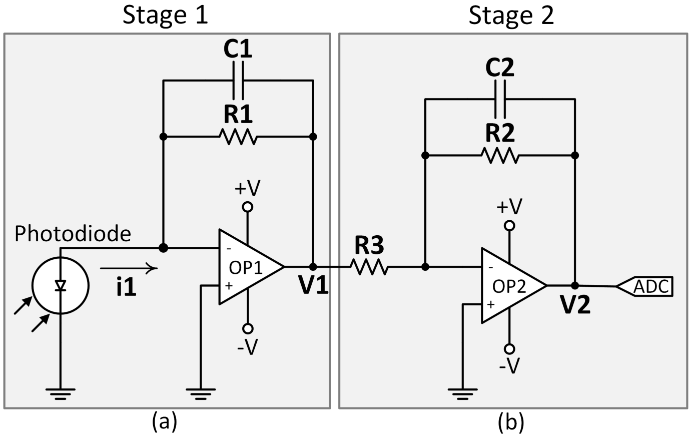


Figure 3.7: Photodiode amplifier circuit: (a) transimpedance amplifier, also known as current-to-voltage converter, and (b) inverting amplifier.

The amplifier circuit is utilized not only to amplify a small signal generated by a photodiode, but also to convert the current into the voltage for an Analog-to-Digital Converter (ADC). The amplifier circuit is composed of two operational amplifier stages. The first stage is a transimpedance amplifier (also called current-to-voltage converter), which is shown in figure 3.7a. Since a photodiode is connected to the inverting input of the operational amplifier, and the input impedance is infinite, the output voltage $V_1$ is the negative product of the photocurrent $i_1$ and the feedback resistor $R_1$; that is $V_1 = -i_1 \times R_1$. The ADC needs to receive positive voltage, hence an inverting amplifier is required in the second stage to convert the negative voltage to a positive voltage. The gain of the inverting amplifier is $\frac{V_2}{V_1} = -\frac{R_2}{R_3}$. The transimpedance amplifier and inverting amplifier circuits also need feedback capacitors in parallel with feedback resistors to reduce noise by limiting the bandwidth [44]. The capacitance can be computed from with the formula $C_1 = \frac{1}{2 \times \pi \times f_{3dB} \times R_1}$, where $f_{3dB}$ is the desire circuit bandwidth.

The ADC pin of a microcontroller is used in the sensor design. The microcontroller has two main functions — data acquisition and sensor network communication. It can acquire four channels of analog signals, which are generated by the photodiodes, amplified by the amplifier circuits, and converted into digital signals in situ. The microcontroller also provides different communication protocols (e.g., $I^2C$, SPI, UART) that enable the sensor node to receive commands or send data in a sensor network system. The 8-bit AVR-based microcontroller – ATtiny84A is employed in this research. To minimize the size of the light sensor, the 14-pin small-outline (SOIC) package is chosen. Most I/O ports of the ATtiny84A microcontroller have alternative functions. The light sensor uses four I/O ports as 10-bit A/D converter and three I/O ports (pin 7 to 9) as a MOSI (Master Out/Slave In) pin, a MISO (Master In/Slave Out) pin, and a clock pin for Serial Peripheral Interface (SPI) communication. Using SPI communication, multiple light sensors can be connected in a daisy chain configuration to form a sensor network.

The finished product of the light sensor shown in figure 3.8 is a 1.57" × 1.57" (4 cm × 4 cm), 4-layer FR-4 printed circuit board (PCB). The top layer and the bottom layer are signal planes. The two middle layers are the ground plane and the power plane. The major electronic components include – a surface mount OPR5911 quad-photodiode, two SOIC LMC660 CMOS quad operational amplifiers, and a SOIC ATtiny84A 8-bit AVR microcontroller. Two important points regarding the PCB design for the light sensor need to be mentioned here: 1) The use of *decoupling capacitors* to stabilize DC current from the power supply [55]. The DC current may contain transient spikes or high-frequency noise. Decoupling capacitors can be placed between the power pins and the ground pins of ICs to shut unwanted current (i.e., AC current) to ground. 2) Digital and analog *ground separation* [28]. The problem known as a *ground loop* is illustrated in figure 3.9, where a photodiode is connected to the negative input of an operational amplifier; besides the photodiode, the positive input of the amplifier, and the DC power supply share a common ground on the circuit board. Mixing the power supply ground, which creates a return path for the DC supply current, and the reference ground, which is a reference point for signal measurement, causes a problem. Because in reality the ground plane is not a perfect conductor, but has a small resistance $R_g$ and inductance $L_g$ (figure 3.9 (b)), when a returning supply current passes the ground path, it produces a voltage drop. Since the power supply ground and the reference ground are connected together, the signal is interfered with by the voltage drop. Moreover, the DC supply current containing high-frequency noise can also cause the signal to be unstable. Hence, the power supply ground and the reference ground are separated in the PCB of the light sensor to break the problematic ground loop. The two grounds are merged at the power supply unit.

### Software Design

The on-board microcontroller (MCU) manages the recording of the measurement, which are then filtered and communicated to a laptop through an Arduino Due.

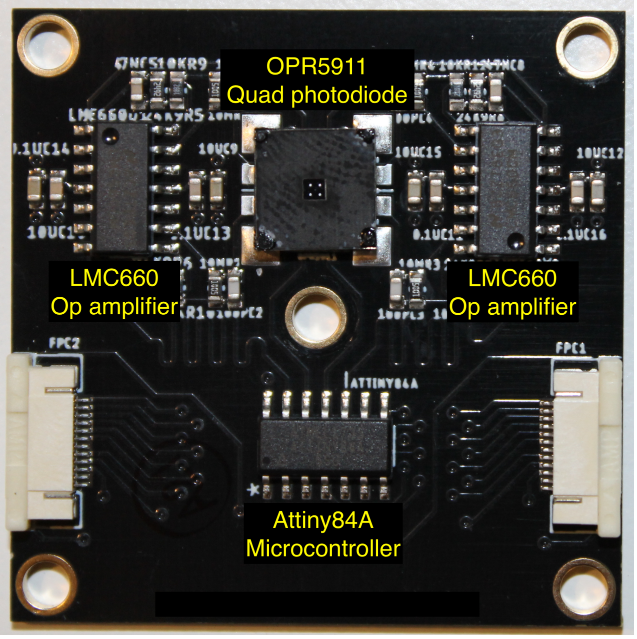


Figure 3.8: The finished PCB of the light sensor.

The on-chip 10-bit analog-to-digital converter (ADC) of the ATtiny84A is used to read the four outputs of the analog amplifiers. To minimize the noise during a readout from the ADC, the MCU goes into a sleep mode disabling the CPU, Flash and I/O clocks. The code also prevents the ADC to make measurements while a communication is on-going as the clock signal strongly affect the readouts.

After reading its value through the ADC, each intensity is filtered by a simple Infinite Impulse Response filter (IIR) [65]:

$$\begin{aligned} I_k(t_n) &= \left(1 - \frac{1}{2^m}\right) I_k(t_{n-1}) + \frac{1}{2^m} ADC_k(t_n) \\ &= I_k(t_{n-1}) + \frac{ADC_k(t_n) - I_k(t_{n-1})}{2^m} \end{aligned} \tag{3.31}$$

where $I_k$ is the intensity of photodiode $k$, $t_n$ is the $nth$-measurement time, $ADC_k(t_n)$ is the ADC readout of channel $k$ at time $t_n$, and $m$ is a positive integer.

This filter is very efficient in terms of computation time and memory space. It does not require floating number calculations as the division by a power of two simply corresponds to a bit-shift operation. It can be used at high frequency (over 1 kHz) and with small MCUs such as the one used in this circuit. Its C++ implementation is:

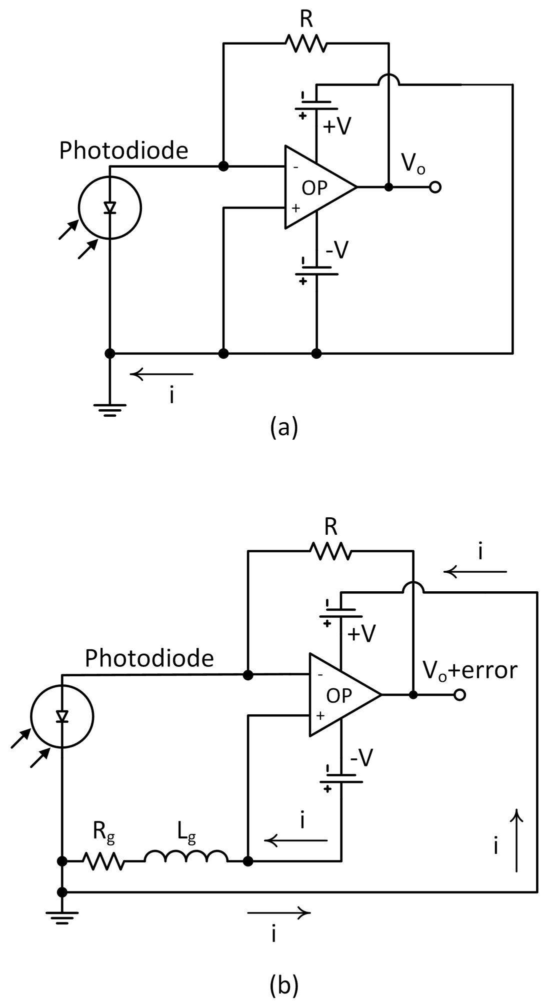


Figure 3.9: Grounding problem: (a) power supply and signal reference share a common ground, (b) a real ground conductor may contain little resistance and inductance; therefore, a returning current passing the ground path creates a voltage drop which can interfere with the reference ground.

```
intensity[k] += (ADC - intensity[k]) >> m;
```

where `ADC` points to the register where the reading of the analog-to-digital converter is stored.

To communicate the filtered intensities, the sensors transfers them to an Arduino Due using the SPI protocol. Contrary to the normal use of SPI which requires each slave to have its own slave select line (see [49]), we only use a single line for all the sensors and differentiate them by an address byte at the beginning of the transmission. SPI was used for its reliability, speed and easy scalability compared to I2C or UART, the other options available on the microcontroller.

## 3.4 Experimental Setup

### Plate with Integrated Sensors

A simple thin plate made out of aluminum was chosen as the surface for the experiment, the goal being to reconstruct its shape using embedded light sensors and the algorithm described in section 3.2. The plate is 0.016" (0.41 mm) thick, 50" (127 cm) long and 10" (25.4 cm) wide. It can be seen in figure 3.10. The relatively small thickness allows the structure to undergo large bending deformations while keeping small in-plane forces guaranteeing the inextensibility of the deformations. It is however thick enough to prevent wrinkling or buckling. Its dimensions were chosen so that it fits on an optical table while still being large enough to accommodate multiple sensors.

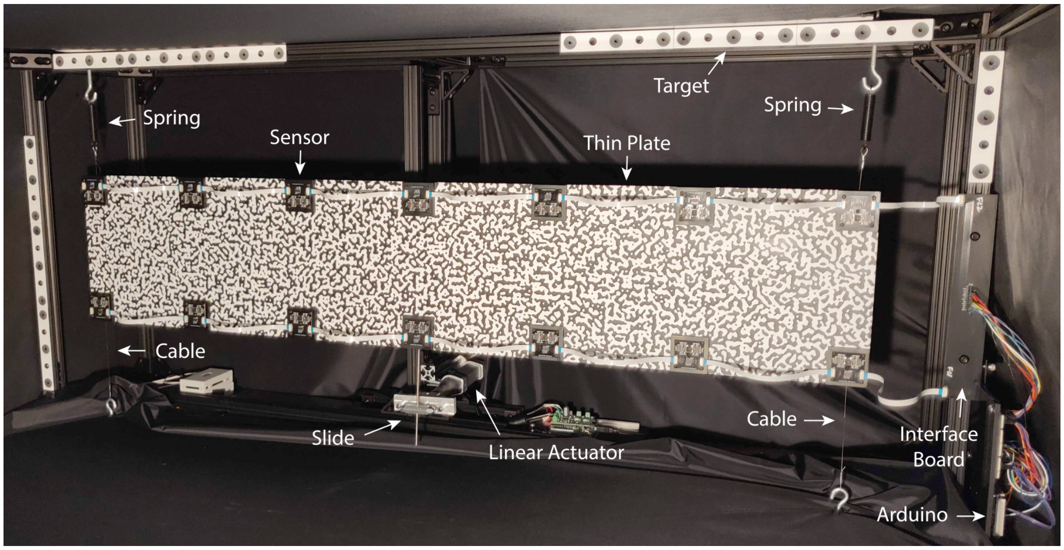


Figure 3.10: Photo of the experiment. The plate with a black and white DIC pattern holds 14 sensors placed on rigid supports. The plate is held by tensioned cables at each end and two linear actuators in the middle (the actuator attached to the center of the structure is not visible). Targets all around the setup are used to define the reference frame.

Sensors were mounted on 2”×2” rigid supports made of aluminum (see figure 3.12). These supports prevent deformation of the circuit boards when the aluminum plate bends, which could damage the solder joints. They also prevent the sensor from moving during the experiment which would introduce errors in the measurements. The center of the support is separated by 8" (20.32 cm) in both directions as seen in figure 3.10.

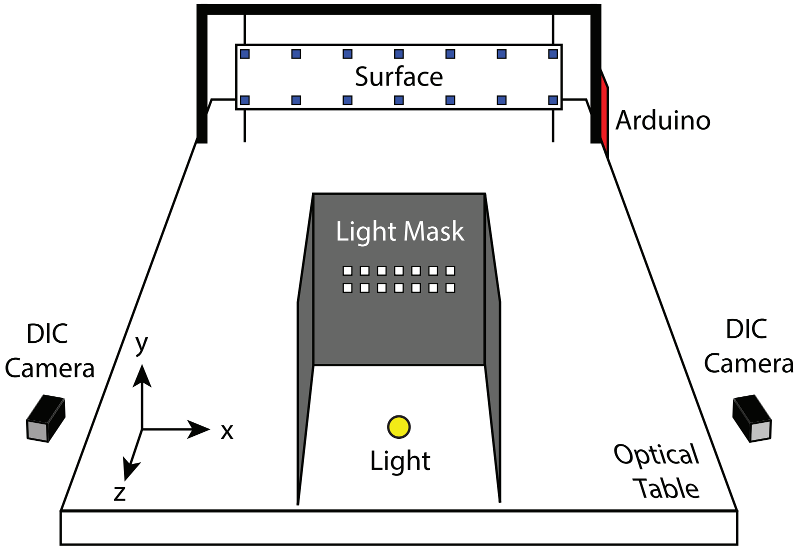


Figure 3.11: Schematic of the setup. The surface with 14 sensors is at one end of an optical table with an Arduino retrieving the measurements, and the LED light is in an enclosed box at the other end. DIC cameras produce a secondary measurement of the shape of the surface.

The structure is held at both ends by tensioned steel cables. Each cable passes through two aluminum supports that are attached at the top and the bottom of the structure (see figure 3.12). The cables are pinched by a set screw in the middle of the top support to prevent the structure from sliding down. These boundary conditions allow rotations and easy bending of the surface and effectively create simply supported boudary conditions. The steel cables were tensioned by a spring to about 35 N.

Deformation of the structure is imposed by two linear actuators placed in the middle (see figure 3.10). The top one, placed at the center of the structure is attached using a pivot while the bottom one is connected using a pivot and a slide mechanism. As a result, when both actuators are actuated together, the structure undergoes pure deflection and when only the bottom one is actuated, the structure undergoes pure torsion. Each actuator contains an encoder to measure its extension. The actuators were calibrated using a laser ranging system (Keyence LK-G157) to achieve an accuracy of 0.5 mm. They are controlled using a P-controller loop programmed in an Arduino Uno.

A total of 14 sensors are placed on the structure. They are all connected in a daisy chain using Flat Flexible Cables (FFC) which have a much lower bending stiffness than the aluminum sheet and do not contribute to its deformation. The sensors

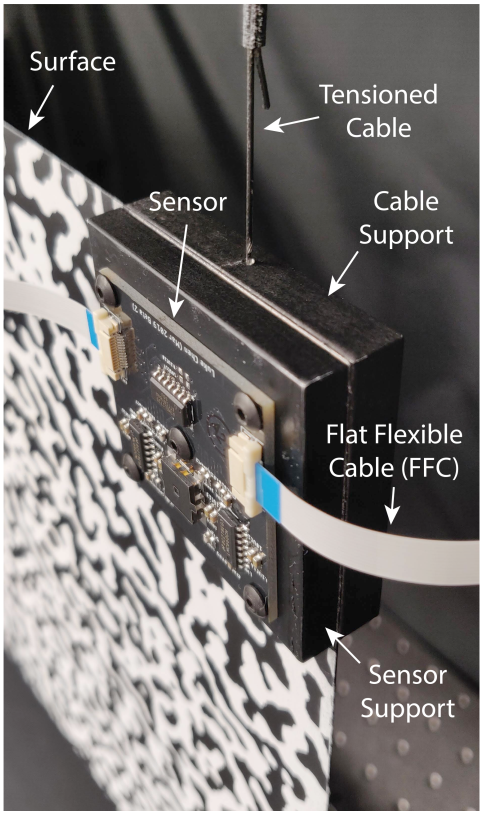


Figure 3.12: Closeup view of the top-right sensor mounted on its rigid support. The aluminum sheet is sandwiched by another support where a tensioned cable passes through and is pinched to hold the structure.

share the same power lines: 3.3V for the on-board microcontroller, +/- 12V for the operational amplifiers, and digital and analog grounds which are provided by a power supply.

The top and bottom rows of sensors connect to an Arduino Due through an interface board fixed to the optical table (see right side of picture 3.10). The Arduino then relays the measurement data to a laptop via Serial communication. The Arduino effectively acts as a custom Serial to SPI converter. A MATLAB-based software was written to retrieve measurements from all sensors as well as changing their filtering parameters.

The light source used in the experiment is a white LED (SSR-90 from Luminus Devices Inc.). It generates about 1000 lumens which is matched with the gains of the sensors to generate a readable output. The light intensity varies along the span of the structure and decreases by about 10% on the edges relative to the center. It is placed about 2 m in front of the structure as shown in figure 3.11. In order to avoid light reflections which could affect the reading of the sensors, the whole setup is covered in blackout fabric (BK5 from ThorLabs). A mask was placed in front of the LED in order for the light to only direct towards the sensors and hence avoiding polluting the setup with unnecessary light (see figure 3.11).

### Secondary Measurement Setup

In order to estimate the performance of the shape reconstruction method, a second measurement system is introduced to measure the shape of the surface. A digital image correlation (DIC) system from Correlated Solution was used. It consists of two cameras (Point Grey Grasshopper3, 5MP, Mono), pointed to the structure from two opposite directions (see figure 3.11). The Vic2D software was used to combine pairs of images taken by both cameras at the same time to triangulate each point on the surface. A calibration is performed to know the position and angles of each camera relative to each other as well as their internal parameters.

A black and white pattern needs to be drawn on the surface before taking any image. The size of the features needs to be more than 3 pixels to avoid aliasing, and it is preferable to have the same amount of black and white in random shapes. A computer generated pattern was created using Perlin Noise [56, 58]. While this noise was initially used in computer-generated imagery (CGI), it has been proven very efficient to generate a random pattern, compatible with a DIC system [56]. A grid of 6 mm was used in the algorithm and a binary filter was applied to the results (all negative pixel values were set to black and positive set to white). The surface was first painted with a flat black spray paint and then was covered with painter's tape cut according to the computer generated pattern on a laser cutter (Universal Laser Sytem XLS10MWH, 75 W 10.6 $\mu m$ $CO_2$). The exposed structure was then painted with a white flat spray paint and the remaining tape was removed exposing the black layer of paint. The result can be seen in figure 3.10.

The error from the DIC system using such pattern was assessed on a rigid 8" × 8" plate. Images of this plate placed at different locations were taken. Rigid-body transformations were removed by software and the remaining displacements between each image were computed. The error was less than 50 $\mu m$ RMS. This number was also assessed on the experimental surface by measuring it with a FaroArm laser scanner. Such system has a similar error as the DIC. By comparing the shapes given by both systems, we were able to estimate a standard deviation in the order of 100 $\mu m$ which agrees with the previous error.

Finally, it is important to notice that the reconstructed shape from the algorithm is located in a reference frame with the light at its origin. In order to place the DIC shape in the same reference frame, the location of the LED light needs to be known. This is the goal of the 30 targets placed around the structure (see figure 3.10). Their position has been precisely measured relative to the LED by a FaroArm. The shape of the structure is then measured in reference to these targets by the DIC system and translated to fit in the reference frame with the light at its origin.

## 3.5 Calibration of the Sensors

### Relative Calibration

A calibration was performed to check the response, performance, and accuracy of the sensors. Different effects can impact equations 3.19 and 3.20 describing the response of a perfect sensor. The gains of the analog circuits may be different from one channel to the other as the resistance values may differ. The aperture placed on top of the sensor may not be perfectly square nor placed exactly in the middle of the photodiode array.

A 2-axis stage was designed using two Newport PR50 rotation stages (figure 3.13). They have a precision of 0.01° which is an order of magnitude lower than the expected precision of the sensors. The rotation stages are placed at a right angle from each other and their rotation axes intersect at the center of the aperture hole of the sensor. The sensor is placed in a slot to guarantee its precise alignment. Sensors were calibrated from $-20°$ to $20°$ in both directions with increments ranging from 0.5° to 2°. A total of 100 measurements (10 sec at 10 Hz) was gathered to average out the noise of the sensor.

Figure 3.14 shows the intensity of the four photodiodes during a calibration with 0.5° increments. The intensities increase as the light spot covers more area of each photodiode. Note that the light source was slightly higher than the sensors hence the light intensity is higher for the bottom photodiodes. Figure 3.15 shows the standard deviation of the intensities of each photodiode calculated from the 100 measurements at each point. The figure shows a circular pattern where the noise is larger for a constant intensity. This pattern appears on all photodiodes for the same intensities.

Figure 3.16 shows the angles $\alpha$ and $\beta$ imposed by the rotation stages as a function of the averaged output of the sensors $R_\alpha$ and $R_\beta$. The color scale shows the difference between the measurement and the expected calibration output, equations 3.22 and

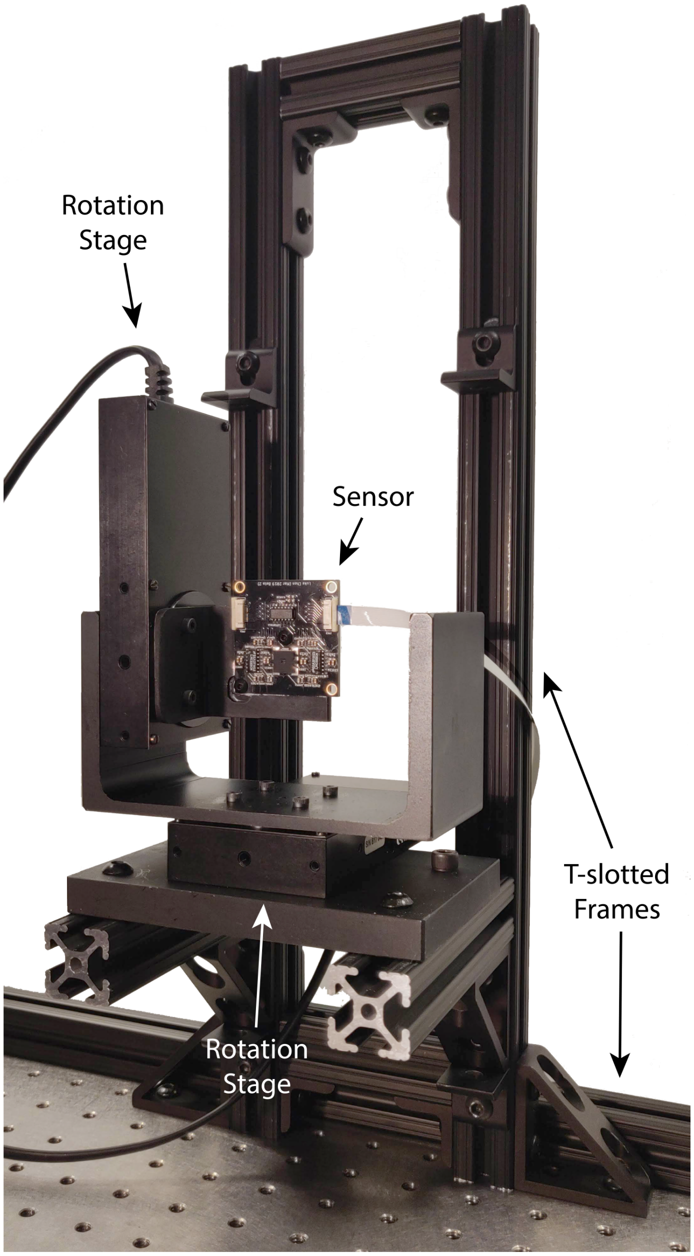


Figure 3.13: Stage used to calibrate the sensors. The T-slotted frames allow the stage to move sideways and up and down.

3.23. The refractive index of the aperture $n$ was left unknown as well as possible shifts of the aperture. These shifts simply add a constant to the numerator of each equation. These equations were fitted to the result which led to a refractive index of 1.56 which is typical for polymers [13] and a shift of less than 50 $\mu m$ in both directions. Overall, the experimental results agree qualitatively and quantitatively well with the theory. The output of the sensors $R_\alpha$ and $R_\beta$ are mainly dependent on

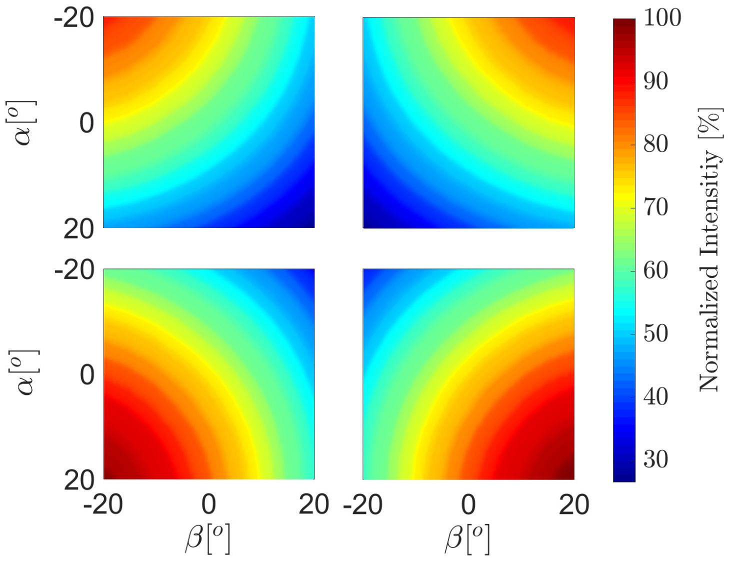


Figure 3.14: Intensity of the photodiodes as function of the angles of the calibration stages.

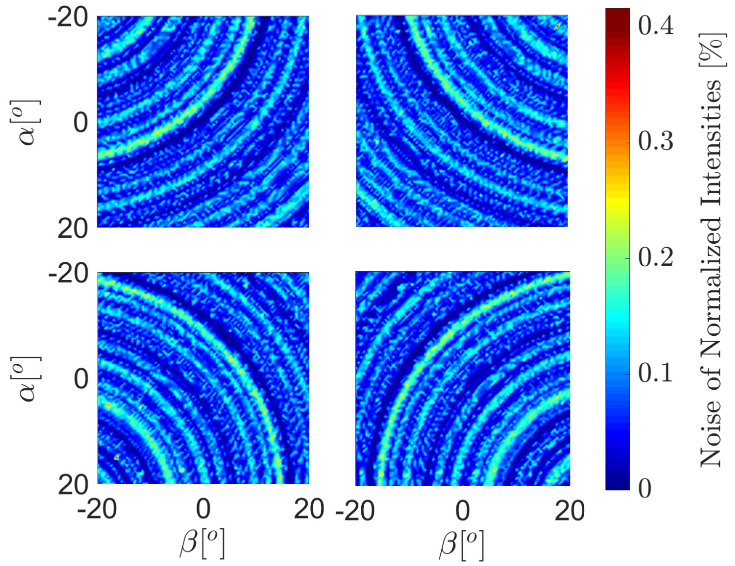
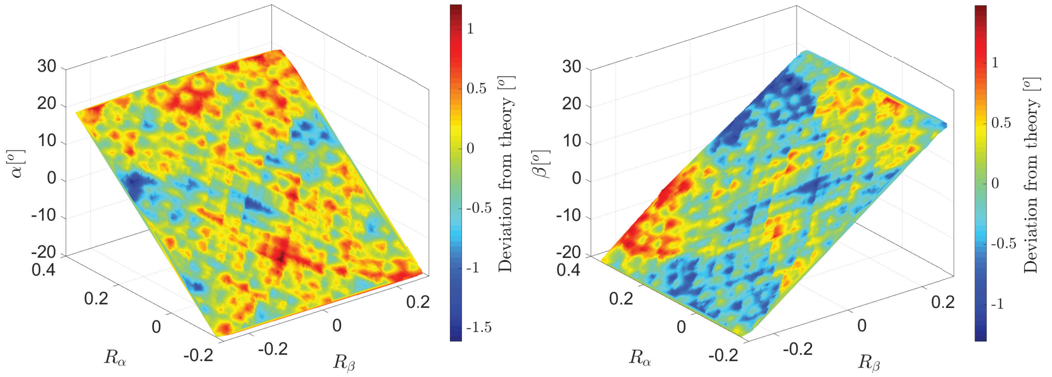


Figure 3.15: Noise of the intensities of each photodiode based on 100 measurements at each point.

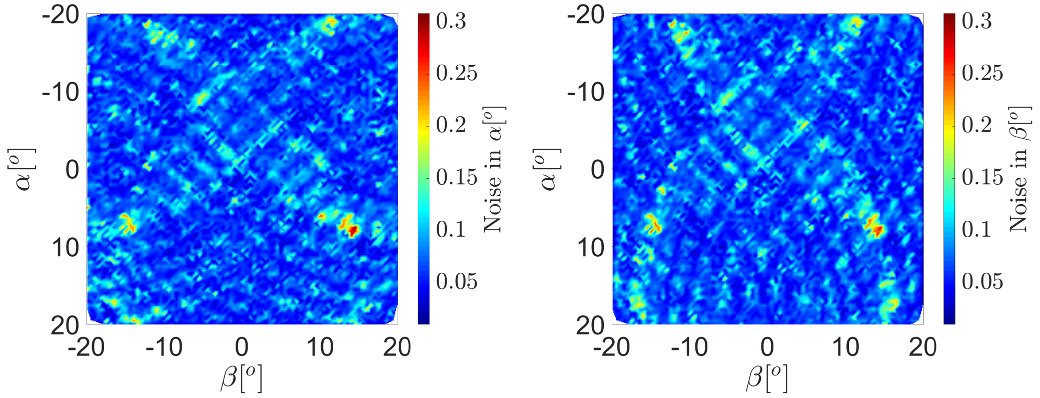
one of the angles, as predicted. The difference between the theory and the calibration can be up to 1.5° and a circular pattern is noticeable in the error. These features mainly correspond to slight gain variations between the photodiodes. These features are few degrees in size, hence a calibration with 2° increments is deemed sufficient to capture them which considerably accelerates the calibration process.



(a) Calibration surface of the angle $\alpha$.

(b) Calibration surface of the angle $\beta$.

Figure 3.16: Calibration surfaces. The colors show the difference in degrees between the calibrated angles and the theoretical response.



(a) Noise in reconstructed angle $\alpha$.

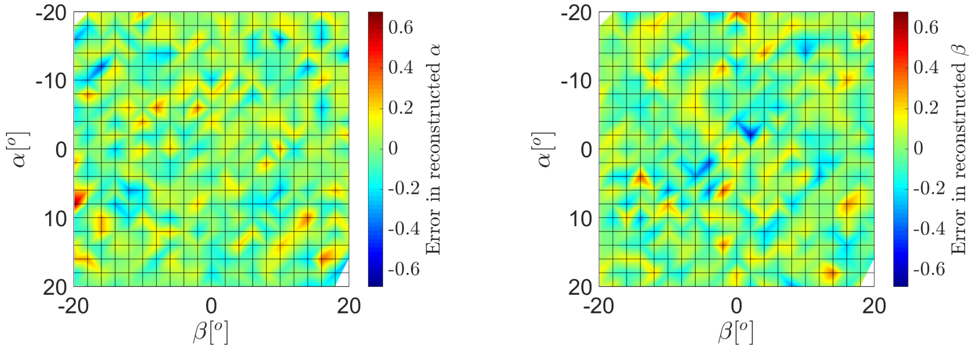
(b) Noise in reconstructed angle $\beta$.

Figure 3.17: Noise of the reconstructed angles from the calibration data.

Using the calibration curves, the noise in intensities (or equivalently in $R_\alpha$ and $R_\beta$) can be converted to noise in $\alpha$ and $\beta$ (figure 3.17). The figure follows the noise pattern of the intensity of the photodiodes with circular features. The peaks in noise correspond to the intersection of the noise circles seen in figure 3.15. On average, the noise in the angles is on the order of 0.1°.

As the error of the calibration curves is an order of magnitude higher than the noise of the sensor and relatively unpredictable, a calibration with 2° increments was performed for all sensors.

The repeatability of the calibration was first tested by conducting two identical calibrations with 2° increments. The difference between the results are shown in figure 3.18. The error is around 0.2° RMS. Note that the output of the sensor is

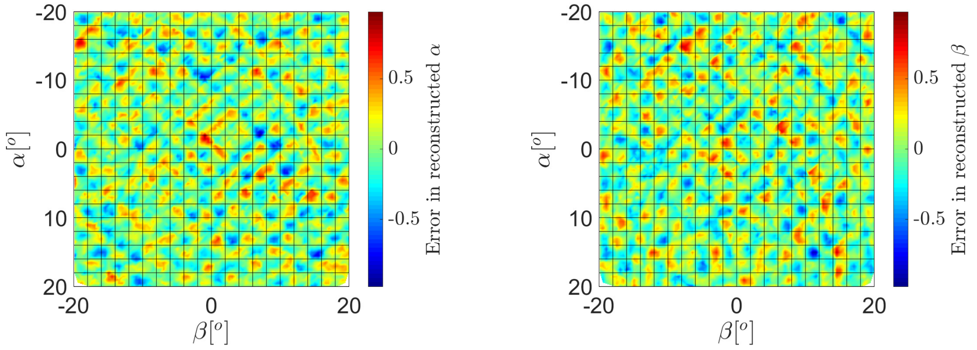


(a) Repeatability error on $\alpha$.

(b) Repeatability error on $\beta$.

Figure 3.18: Repeatability error between two 2° calibrations.

averaged from 100 measurements at each step, effectively suppressing the noise of the sensor. This error is possibly due to variations in the response of the sensor from thermal effects, electronics noise or variations in the light intensity. These repeatability errors are on the order of the noise of the sensors, hence no further work was conducted to try to eliminate them.

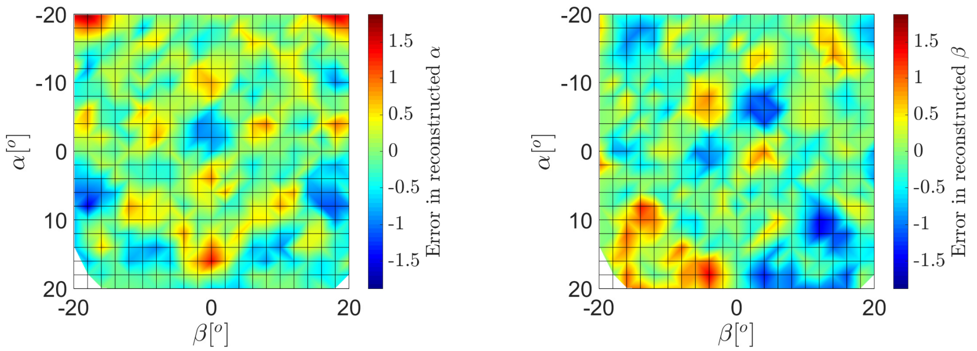


(a) Accuracy on $\alpha$.

(b) Accuracy on $\beta$.

Figure 3.19: Accuracy of a 2° calibration using a 0.5° calibration as reference.

The accuracy of the 2° calibration is verified by comparing it with the 0.5° calibration shown in figure 3.16. The 2° calibration is fitted with a 2D linear interpolation method (*linearinterp* option in the function *fit* in MATLAB). The fit is used to predict the angles at each step of the 0.5° calibration using the output of the sensor only. This result is compared to the actual angles imposed by the rotation stages during the 0.5° calibration. Figure 3.19 shows the results of that comparison. The angle error is 0.3° RMS. Part of this error can be due to the stability issues mentioned above. Finer calibrations (1° increments for instance) did not improve the accuracy significantly.



(a) Error in $\alpha$

(b) Error in $\beta$.

Figure 3.20: Error between two 2° calibrations under different light intensities.

It has been shown in section 3.3 that the centroid of the light spot $R_\alpha$ and $R_\beta$ is independent of the incoming light intensity. To prove this statement, another calibration with 2° increments, but with a higher light intensity was performed and compared to the original one. The intensity was increased by about 50% by increasing the voltage across the LED light source. Errors between the two calibrations are shown in figure 3.20. We observe errors much larger than the previously mentioned ones. As a result, the output of the sensor is dependent on the light intensity. This is due to different sensitivities and gain between the photodiodes and their respective electronics. As a result, some terms do not cancel out in the calculation of the spot centroids (equations 3.17 and 3.18).

In order to limit the errors due to different light intensities across the surface, the sensors were calibrated near their position during the experiment.

### Bias Calibration

In the algorithm described in section 3.2, a sensor should measure 0° if the surface on which it is mounted is locally perpendicular to the incoming light ray. This is, however, not the case for the stage-calibrated sensors.

The first reason is related to the alignment of the calibration stage during the calibration process. While each sensor was calibrated near its position during the experiment, the stage was always pointing along the Z-axis when set to 0° in both directions (the coordinate system for the experiment can be seen in figure 3.11). There is a geometric offset between the Z-direction and the direction of the light during calibration. It can be corrected by measuring the position of the sensor and the light source in the reference frame of the experiment and subtracting the calculated angles from the calibration data.

In addition to this effect, the sensors were not carefully aligned with the structure when they were mounted on it. There may have been some tip and tilt due to uneven paint thickness, machining tolerance, etc. A similar effect happens when mounting the sensor on the calibration stage (which was furthermore not perfectly aligned to the Z-axis). All these small effects can add up to create a bias error in the measurement of each sensor. As shown in the previous sub-section, even an offset of 0.5° is larger than the accuracy of the sensors.

To circumvent this effect, the surface was measured using a FaroArm laser scanner. It was then possible to estimate the local slope of the structure around each sensor in the reference frame of the experiment by performing a local linear fit around each sensor. Different sizes of windows were considered and their influence was investigated. Errors on the order of 0.5° between window sizes were observed. To further gain confidence about the actual angle seen by the sensors, the reconstruction algorithm described in section 3.2 was applied, solving for the sensor angles that minimize the RMS error between the reconstructed surface and the point cloud of the surface generated by the FaroArm. The algorithm parameters displayed in figure 3.2 were used ($5 \times 3$ grid of control points and $13 \times 5$ inextensibility grid). The results are shown in figure 3.21. The RMS error between the algorithm and the point cloud converged to 0.2 mm.

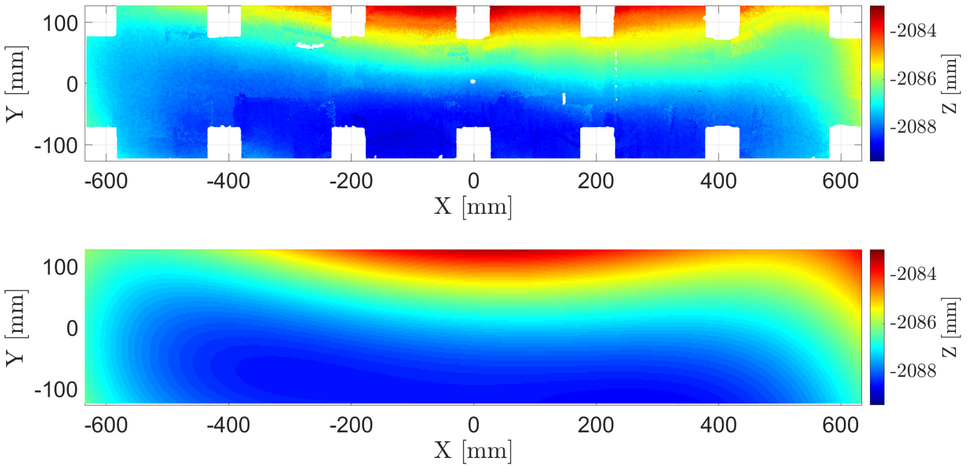


Figure 3.21: FaroArm shape (top) and optimally reconstructed shape (bottom) that was used to compute the bias of each sensor.

## 3.6 Experimental Results

Multiple experiments were conducted to show the performance and accuracy of the surface reconstruction algorithm described in section 3.2 using the light sensors presented in section 3.3 on the structure introduced in section 3.4. All the sensors were calibrated and corrected for their bias, as shown in section 3.5, prior to the experiments.

During each experiment, the structure remained static. The calibrated DIC system was used to take photos of the surface with the two cameras to estimate the shape of the structure with great accuracy. A total of 1000 measurements were taken with the sensors (100 secs @ 10Hz) which were converted into 1000 separate reconstructed surfaces. Such a large number of measurements gives statistically accurate results.

Fifteen control points were used in the reconstruction algorithm. They were uniformly spaced on a $5 \times 3$ grid as shown in figure 3.2. An inextensibility grid of $13 \times 5$ was used. The modifications of the algorithm presented in section 3.2 were implemented. Because the surface was located relatively far from the light (about 2 m), the algorithm was not able to accurately estimate the distance, which could have lead to large errors in the reconstruction. To prevent what is a numerical singularity of the algorithm (see chapter 2), the central point of the surface was fixed to its 3D position extracted from the DIC data. This is explained in more detail in chapter 2.

The two actuators placed in the middle of the structure applied bending and torsional deformations. If they were both moved together, the surface undergoes “pure” bending. Torsion can be applied by moving the bottom actuator relative to the top one. The performance of the algorithm was studied for different amplitudes of bending and torsion.

### Bending of the Surface

The structure was deflected at its center by applying the same motion to both actuators.

Figure 3.22 shows the comparison between the DIC shape (on top) and the mean result of the reconstruction algorithm (at the bottom) after bending the structure by 20 mm (both actuators were driven 20 mm forward). The mean was calculated by averaging the 1000 reconstructed shapes. The DIC can only measure the shape of the central part of the structure as the cable connecting the sensors obstructs the view of the cameras. Figure 3.23 shows the difference between the shapes. The error varies along the surface in range ±1 mm. It is 0 at the center since this point is constrained to be equal to its position measured by the DIC.

The RMS error of each reconstructed shape can be calculated and a histogram of the error can be plotted (figure 3.24). The histogram was accurately fitted with a Generalized Extreme Value distribution. The mean error is around 0.7 mm.

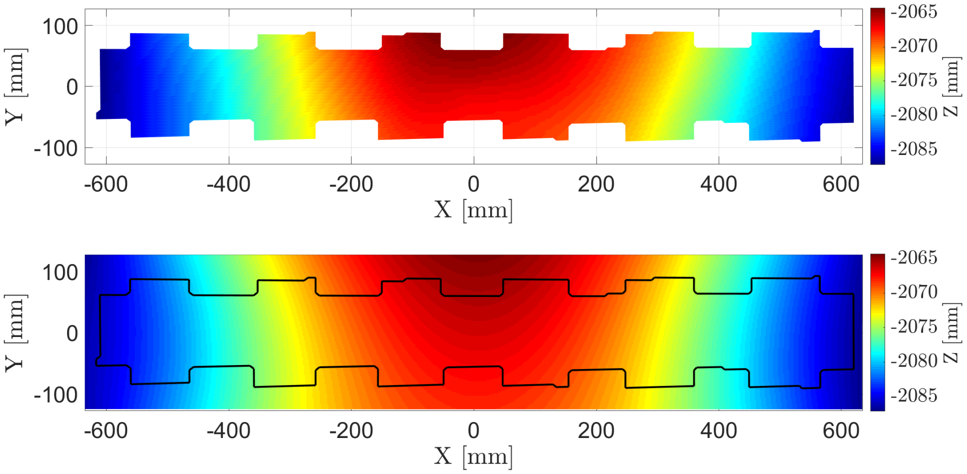


Figure 3.22: DIC shape (top) and mean reconstructed shape from the sensor data (bottom) for a 20 mm bending of the structure. The black outline represents the boundaries of the DIC result to ease comparison.

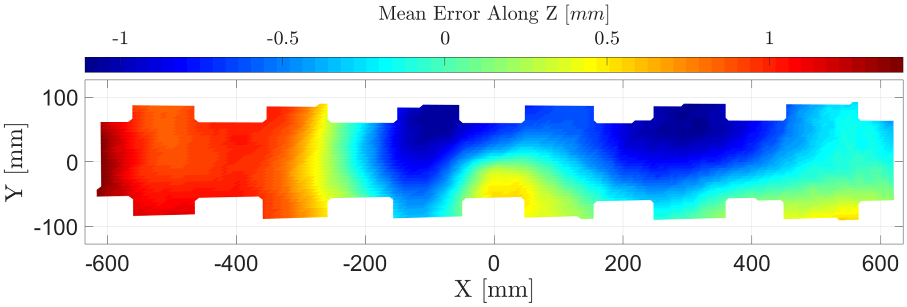


Figure 3.23: Difference between the DIC shape and the mean reconstructed shape from the sensor data for a 20 mm bending of the structure.

### Torsion of the Surface

Figure 3.25 shows a similar reconstruction with the structure undergoing “pure” torsion. The bottom actuator was moved by 20 mm forward while the middle actuator remained at its 0 position. This imposes a 5.2° rotation between the actuators. Because of the compliance of the aluminum plate, only the bottom part

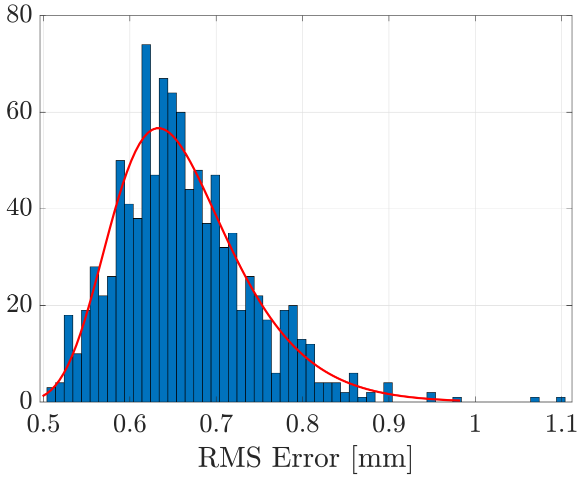


Figure 3.24: Histogram of the RMS error between the 1000 reconstructed shapes and the DIC result for a 20 mm bending of the structure.

deforms while the top remains mostly flat. The figure shows that the algorithm is capable of qualitatively reconstructing this more complex shape. The error shown in figure 3.26 is concentrated in the central region that deforms the most and where localized deformation happens. On average, the RMS error is about 0.8 mm as shown in the histogram figure 3.27. The Generalized Extreme Value distribution is again suited to estimate the performance of the system.

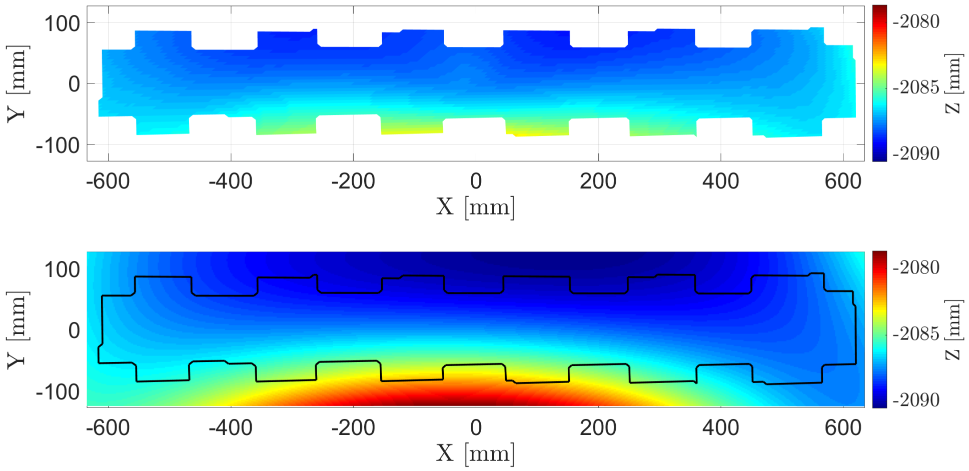


Figure 3.25: DIC shape (top) and mean reconstructed shape from the sensor data (bottom) for a 5.2° torsion of the structure. The black outline represents the boundaries of the DIC result to ease comparison.

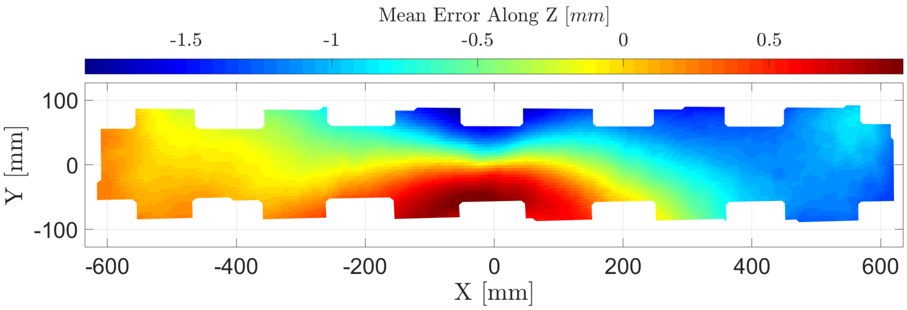


Figure 3.26: Difference between the DIC shape and the mean reconstructed shape from the sensor data for a 5.2° torsion of the structure.

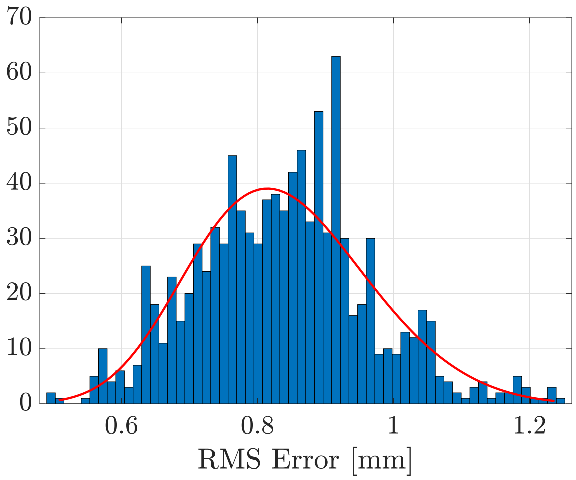


Figure 3.27: Histogram of the RMS error between the 1000 reconstructed shapes and the DIC result for a 5.2° torsion of the structure.

### Combination of Deflection and Torsion

To further understand the evolution of the mean RMS error of the reconstruction for different deformations, the deflection was varied from -20 mm to 20 mm and the torsion from $-5.2°$ to $5.2°$. The deflection of the surface (in millimeters) is defined by the position of the central actuator, while the torsion (in degrees) is defined by the angle between the actuators. For all cases, 100 measurements were recorded from the sensors leading to 100 reconstructed shapes that are then compared to the DIC.

The mean RMS error across the surface is plotted against the imposed deflection and torsion, figure 3.28. The minimum error happens for the flat case (no bending. nor torsion) which is simply due to the fact that the bias of the sensor was calibrated

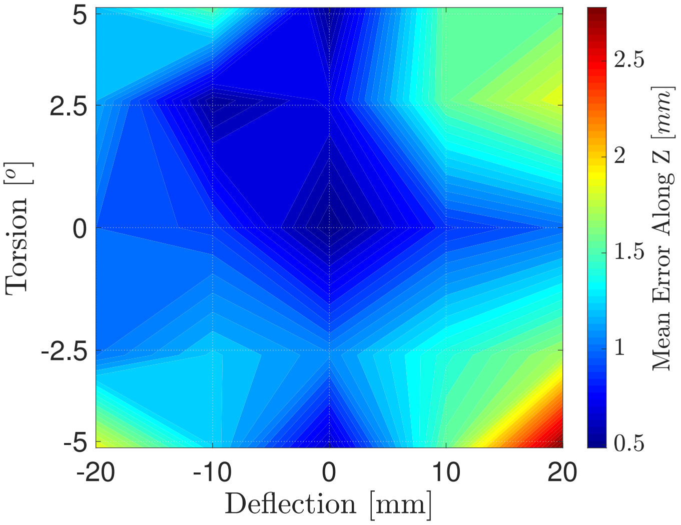


Figure 3.28: Mean RMS error of the reconstructed shapes for different deflection and torsion of structure.

to minimize the RMS error of this shape. The error is on the order of 1 mm for small deformations, but increases for larger deformations. This can be explained by the increased complexity of the shape of the surface for large actuation. More local features appear which cannot be reconstructed by the algorithm. Figure 3.29, for instance, shows the DIC and reconstructed shapes of the bottom right load case of figure 3.28 (Deflection of 20 mm and Torsion of $-5.2°$). While the overall shape with displacements over 25 mm is reconstructed, local features appear around the center of the structure which leads to errors of up to 5 mm (figure 3.30). The shape is eventually recovered at the edges. Other deformed shapes are very well reconstructed such as 5.2° pure torsion which has a mean error of 0.5 mm. Figure 3.31 shows the superposition of the DIC shape over the mean reconstructed surface. The difference between the shapes is hard to notice as both agree very well with each other.

## 3.7 Conclusion and Discussion

In this chapter, we have demonstrated the accuracy of an algorithm that reconstructs the shape of a surface with embedded light sensors. An experiment was designed to validate the performance of a real system and understand the practical limitations of the proposed solution.

Light sensors were designed to measure the two relative angles between the sensor and a light source. Their simple architecture recreates a pinhole camera using

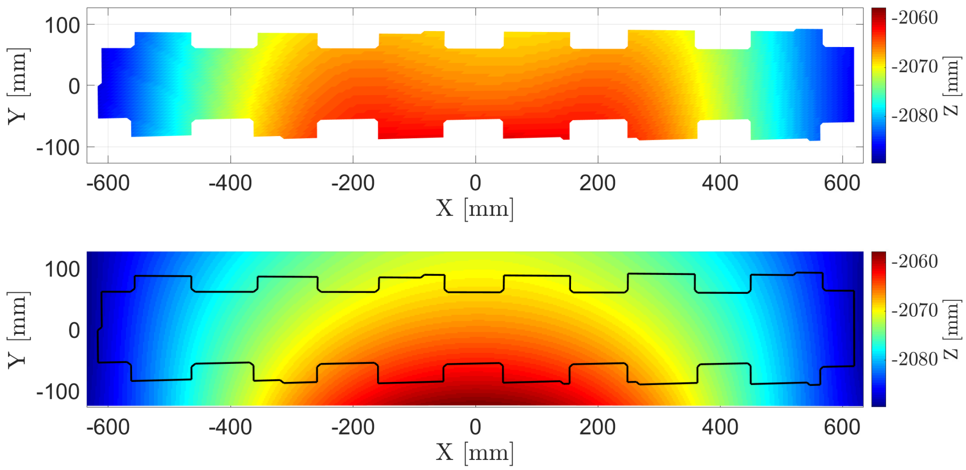


Figure 3.29: DIC shape (top) and mean reconstructed shape from the sensor data (bottom) for a 20 mm deflection and −5.2° torsion of the structure. The black outline represents the boundaries of the DIC result to ease comparison.

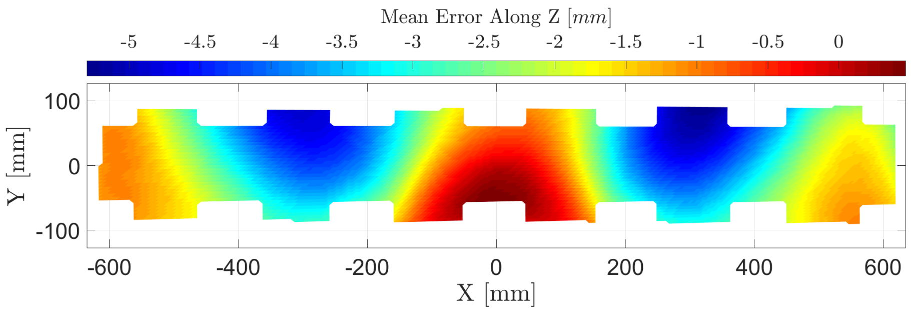


Figure 3.30: Difference between the DIC shape and the mean reconstructed shape from the sensor data for a 20 mm deflection and −5.2° torsion of the structure.

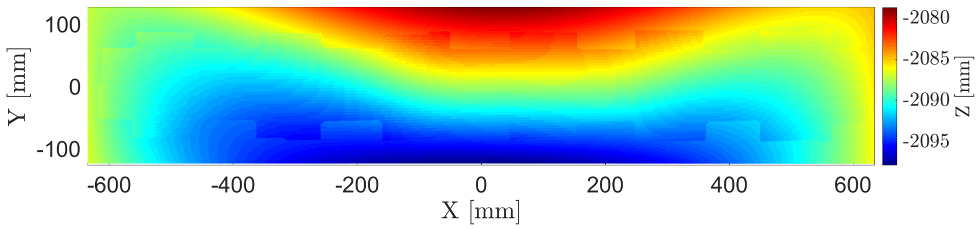


Figure 3.31: Superposition of the DIC shape and the mean reconstructed shape from the sensor data for a 5.2° “pure” torsion of the structure.

a quad-photodiode. They were developed internally in order to understand their functioning, and tailor them to the need of the experiment. The accuracy of the sensors is mostly limited by the noise which varies depending on the angles and the non-linearity of the response. Their accuracy is about 0.5° after calibration, including calibration errors and noise. The calibration process remains complicated and time consuming. Further work is needed to improve their performance, increase their linearity, while simplifying their calibration technique.

The algorithm was tested on a 1.3 m × 0.25 m aluminum thin plate with 14 embedded sensors. It was deformed at the center by up to two centimeters in a combination of bending and torsion. It was shown that this system can reconstruct the shape of the plate with an accuracy of 0.5 mm RMS for the case of simple deflection and torsion and about 1 mm RMS on average for all the experiments that were conducted. For more complex deformations, local deformations of the surface cannot be captured accurately by this algorithm, as it can only compute smooth surfaces. It is important to keep this limitation in mind while developing such a system. Structural considerations need to be investigated to guarantee the relative smoothness of the structure while it deforms in order to accurately reconstruct its shape. This will be improved with a new algorithm in the next chapter.

*Chapter 4*

# FINITE ELEMENT BASED ALGORITHM TO RECONSTRUCT THE SHAPE OF A SURFACE FROM DISTRIBUTED ANGLE MEASUREMENTS

## 4.1 Introduction

The previous chapter demonstrated a system of distributed angle measurement sensors to reconstruct the shape of a relatively large structure. The $1.3 \times 0.25$ m$^2$ aluminum sheet with 14 embedded light sensors was reconstructed to an accuracy ranging from 0.5 mm for simple deformations up to 2.5 mm for complex shapes. Relatively inexpensive and simple sensors were calibrated and used in the experiment. Their precision of 0.5° is average compared to existing light sensor technology [19, 23, 25, 61, 91], but it is sufficient to precisely reconstruct the overall shape of the test structure.

The algorithm is limited by the complexity of the shape of the structure. It was shown that more complex shapes create local deformations that were not fully captured by the algorithm. This is mostly due to the relative smoothness of the shape functions (Lagrange polynomials in this case) and the small number of sensors which limits the resolution of the system.

The algorithm has the advantage of being purely based on geometric considerations, making it easy to be applied to many different applications (as long as the deformation of the surface remain inextensible). Even if some parameters can be tuned, as seen in chapter 2, the simplicity and smoothness of the reconstruction can limit the accuracy of the system as demonstrated experimentally (chapter 3).

In order to more accurately represent the deformation of the structure, especially local deformations, a structural mechanics model is used instead of geometric considerations. Specifically, a finite element (FE) model is designed which predicts the shape of the structure from boundary conditions. While many singularities appear from the geometric conditions (inextensibility equations of the algorithm, see chapter 2), the FE model is well-posed as long as the boundary conditions are well defined.

This approach is inspired from the literature on finite-element modeling which can accurately estimate the shape of a structure from a small number of boundary conditions, effectively creating an accurate representation of the shape from a small number of inputs. This seems appropriate for our problem that uses a relatively small number of sensors.

This approach will be used to generate a new algorithm where the previous inextensibility conditions are now replaced by the state equation of an accurate finite element model of the structure. This new set of equations, together with the measurement equations of the sensors, generates a new, finite element based algorithm. It will be tested on the same experiment described in the previous chapter and the results will be compared with the initial algorithm.

Section 4.2 details the formulation of the finite element based algorithm. It shows how the state and measurement equations are combined to generate a well-posed algorithm.

The details of the finite element model are presented in section 4.3. A small-deformation, plate bending model can accurately describe the deformation of the structures considered for this problem. The model generates a stiffness matrix based on Discrete Kirchoff Elements.

Finally, the new algorithm has been applied to the experiment presented in chapter 3. Results show a net increase in performance over the whole range of deformations undergone by the experimental structure. The accuracy improvements are especially noticeable for complex shapes: the relatively local deformations of the structure, missed by the geometric algorithm are now well taken into account. Little correlation is now present between the complexity of the shape and the accuracy of the method.

## 4.2 Finite Element Based Algorithm

In order to increase the accuracy of the solution, the inextensibility conditions of the algorithm presented in chapter 3 have been replaced with the static state equation generated from a finite element model. The parametrization of the deformation of the shape is also generated from this finite element formulation, following a very similar approach to that presented in chapter 2.

The study is limited to small deformations, plate-like structures like the one used in the experiment shown in chapter 3. A static, linear finite element model using plate elements is designed to capture the behavior of the structure.

The shape of the surface is defined locally in the $(x, y, z)$ reference frame (see figure

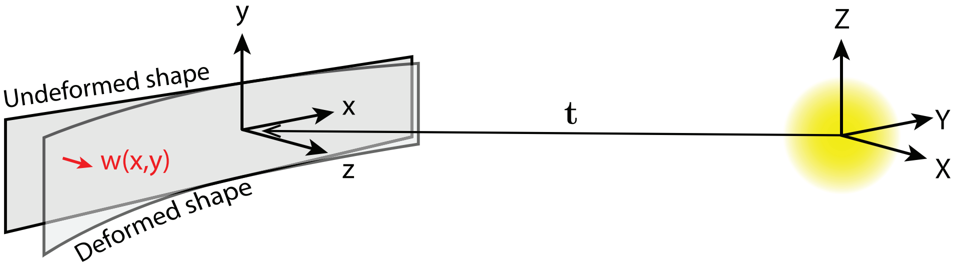


Figure 4.1: Definition of the reference frames of the problem. One has the light at its origin and the other one is used to define the FE model of the structure.

4.1). It is initially planar in the x-y plane. The shape of the deformed surface is defined by the degrees of freedom of the nodes of the finite element mesh. Triangular elements connect all the nodes and shape functions are defined locally on each element to interpolate the shape from the nodal degrees of freedom. For a plate bending model, by neglecting the transverse shear deformation (Kirchhoff plate theory), the degrees of freedom of a node are the out-of-plane displacement and the rotation components of the node in both directions:

$$\mathbf{q}_{fi} = \begin{bmatrix} w_i \\ w,_{xi} \\ w,_{yi} \end{bmatrix} \tag{4.1}$$

where $w$ is the out-of-plane deflection of the surface, $w,_x = \frac{\partial w}{\partial x}$, and $w,_y = \frac{\partial w}{\partial y}$.

The reconstructed surface is translated to fit in a reference frame with the light source at its origin in order to use the measurement equations in chapter 2: equations 2.24 and 2.25 (see figure 4.1). The position of a point on the surface in the light reference frame $(X, Y, Z)$ is:

$$\mathbf{r}(x, y) = \mathbf{t} + \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{N}(x, y) \end{bmatrix} \mathbf{q}_f \tag{4.2}$$

where $\mathbf{t}$ is a translation vector, $\mathbf{N}(x, y)$ is the row vector of the shape functions evaluated at $(x, y)$ and $\mathbf{q}_f$ is the vector of all degrees of freedom of the model. Note that since the problem is invariant for any rotation around the light, no rotation matrix is necessary.

Kinematic boundary conditions are necessary to ensure small deformations in the $(x, y, z)$ reference frame, that is, removing any possible rigid-body motion. Clamping a node (all its degrees of freedom are set to 0) is an example of a sufficient condition. Restraining the out-of-plane deflection of 3 non-aligned nodes to 0 is another example.

A state equation links the displacement components for all degrees of freedom to any external loads in a linear system:

$$\mathbf{K}\mathbf{q}_f = \mathbf{F} \tag{4.3}$$

where $\mathbf{K}$ is the stiffness matrix of the FE model, $\mathbf{q}_f$ is the vector of all the degrees of freedom, and $\mathbf{F}$ is the vector of external nodal loads (interpolation of the external loads to the nodal d.o.f.).

Both kinematic and load boundary conditions can be applied to the model (see [21]). Without loss of generality, only loads are applied as boundary conditions (other than the kinematic b.c. used to remove rigid motions). Only concentrated loads are studied in this chapter. The location of the concentrated loads is assumed to be known, but not their amplitude which will be left as an unknown and solved by the angle measurements of the sensors. Let $N$ be the total number of external loads (forces or moments). The force vector $\mathbf{F}$ can be re-written as:

$$\mathbf{F} = \mathbf{F}_d\tilde{\mathbf{F}} \tag{4.4}$$

where $\tilde{\mathbf{F}}$ is a vector of dimension $N$ concatenating the amplitude of the external concentrated loads and $\mathbf{F}_d$ is a known conversion matrix that redistributes the loads to their nodal values. From equations 4.2, 4.3, and 4.4, one can summarize the unknowns of the algorithm as:

1. the degrees of freedom of the nodes
2. the amplitudes of the forces
3. the translation of the shape

which can be written as:

$$\mathbf{q} = [\mathbf{q}_f^T \quad \tilde{\mathbf{F}}^T \quad \mathbf{t}^T]^T \tag{4.5}$$

The dimension of this vector can be large for dense meshes but it is determined by an equally large number of equations.

The FE model in equation 4.3 replaces the inextensibility equations defined in the original algorithm in chapter 2. The equations for the algorithm are therefore:

1. the static equation of the FE model 4.3:

$$\mathbf{K}\mathbf{q}_f = \mathbf{F}_d\tilde{\mathbf{F}} \tag{4.6}$$

2. the measurement equations which can be written as:

$$\tan\alpha_S = \frac{\boldsymbol{r}(x_S, y_S)\cdot\frac{\partial\boldsymbol{r}}{\partial y}(x_S, y_S)}{\boldsymbol{r}(x_S, y_S)\cdot\boldsymbol{n}(x_S, y_S)}\frac{\|\boldsymbol{n}(x_S, y_S)\|}{\left\|\frac{\partial\boldsymbol{r}}{\partial y}(x_S, y_S)\right\|} \tag{4.7}$$

$$\tan\beta_S = -\frac{\boldsymbol{r}(x_S, y_S)\cdot\frac{\partial\boldsymbol{r}}{\partial x}(x_S, y_S)}{\boldsymbol{r}(x_S, y_S)\cdot\boldsymbol{n}(x_S, y_S)}\frac{\|\boldsymbol{n}(x_S, y_S)\|}{\left\|\frac{\partial\boldsymbol{r}}{\partial x}(x_S, y_S)\right\|} \tag{4.8}$$

   where $\boldsymbol{r}(x_S, y_S)$ is obtained from equation 4.2. The angle $\alpha_S$ corresponds to the angle around $\frac{\partial\boldsymbol{r}}{\partial y}$ and $\beta_S$ is the angle around $\frac{\partial\boldsymbol{r}}{\partial x}$ (see figure 2.7 of chapter 2).

   Note that if the sensors are attached to the nodes of the FE model, the local tangent and normal vectors are expressed from the degrees of freedom:

$$\frac{\partial\boldsymbol{r}}{\partial x}(x_S, y_S) = \begin{bmatrix}1\\0\\w,_{xS}\end{bmatrix} \qquad \frac{\partial\boldsymbol{r}}{\partial y}(x_S, y_S) = \begin{bmatrix}0\\1\\w,_{yS}\end{bmatrix} \qquad \boldsymbol{n}(x_S, y_S) = \begin{bmatrix}-w,_{xS}\\-w,_{yS}\\1\end{bmatrix} \tag{4.9}$$

   which are all unit vectors in a first order approximation.

Because equation 4.6 generates as many equations as the number of degrees of freedom of the FE model, the number of measurement equations needs to be at least greater than the number of unknown loads $N$ plus 3 (size of the translation vector).

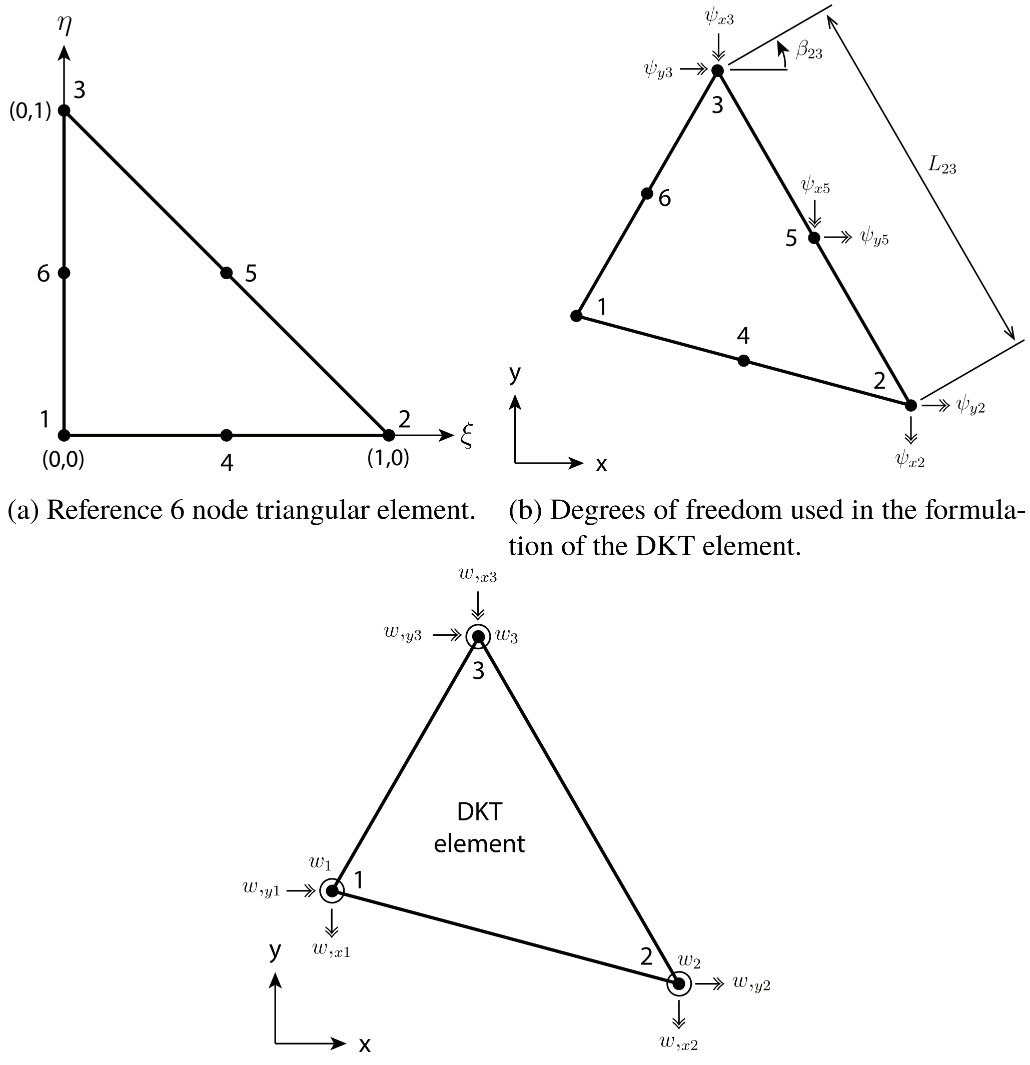
**Solution Method.** The equations 4.6, 4.7, and 4.8 create a system of equations whose unknown is the vector $\mathbf{q}$ (equation 4.5). This system is often overconstrained and is solved using the Levenberg-Marquardt algorithm.

## 4.3 Stiffness Matrix

Discrete Kirchoff Triangular Elements (DKT) are used to generate the linear stiffness matrix in equation 4.6. They have been reliably one of the best types of Kirchoff plate elements for decades [6, 21]. A complete, detailed formulation is presented in [21]. A summary is given in this section.

DKT elements resemble Mindlin elements where the transverse shears are accounted for in the degrees of freedom. The DKT element, however, explicitly enforces zero transverse shear strains at selected locations. As a result, shear strains are not strictly equal to 0 across the surface as stated in Kirchoff theory, but only at discrete points. This relaxation allows the use of complete polynomial fields which guarantees continuity of the deflection and rotations across the structure.

The formulation starts by assuming independent fields for out-of-plane displacements and rotations of a midsurface normal line. In this formulation, the strains of the element are fully calculated by the rotations and the field for the out-of-plane displacement is not explicitly introduced. The triangular element is straight-sided with vertex and midside nodes (see figure 4.2).



(a) Reference 6 node triangular element.

(b) Degrees of freedom used in the formulation of the DKT element.

(c) Final DKT element.

Figure 4.2: Definitions of the degrees of freedom used in the formulation of the DKT element.

Rotation components of the midsurface-normal line are given by complete quadratic polynomials, interpolated from the 6 nodal components:

$$\psi_x = \sum_{i=1}^{6} N_i \psi_{xi} \qquad \text{and} \qquad \psi_y = \sum_{i=1}^{6} N_i \psi_{yi} \tag{4.10}$$

where the shape functions are defined by:

$$\begin{array}{lll} N_1 = (1-\xi-\eta)(1-2\xi-2\eta) & N_2 = \xi(2\xi-1) & N_3 = \eta(2\eta-1) \\ N_4 = 4\xi(1-\xi-\eta) & N_5 = 4\xi\eta & N_6 = 4\eta(1-\xi-\eta) \end{array} \tag{4.11}$$

with $\xi$ and $\eta$ the coordinates of the reference triangular element (see figure 4.2a)

These additional 12 degrees of freedom ($\psi_{xi}$ and $\psi_{yi}$ for $i = 1..6$) are constrained by 12 equations:

1. the first six equations impose zero shear strains at the vertex of the element:

$$\psi_{xi} = w,_{xi} \qquad \text{and} \qquad \psi_{xi} = w,_{yi} \qquad \text{for} \qquad i = 1, 2, 3 \tag{4.12}$$

2. the next 3 equations impose that the transverse shear strain is zero at the midsides.

$$\psi_{si} = w,_{si} \qquad \text{for} \qquad i = 4, 5, 6 \tag{4.13}$$

   where $w_{,si}$ is extrapolated from the out-of-plane displacement and transverse rotations at each vertex. For instance, the transverse slope at node 5 is:

$$w,_{s5} = \frac{3}{2L_{23}}(w_3 - w_2) - \frac{1}{4}(w,_{s2} + w,_{s3}) \tag{4.14}$$

   with $w,_s = -w,_x \sin\beta_{23} + w,_y \cos\beta_{23}$, $\beta_{23}$ the angle between the normal of the edge between nodes 2 and 3 and the x-axis, and $L_{23}$ the length between nodes 2 and 3 (see figure 4.2b). The transverse rotation is calculated identically $\psi_{s5} = -\psi_{x5}\sin\beta_{23} + \psi_{y5}\cos\beta_{23}$.

3. finally, the last three equations impose the normal shear strain to be 0 at the midsides, assuming linearity of the normal slopes between vertices:

$$\psi_{n4} = \frac{1}{2}(w,_{n1} + w,_{n2}) \qquad \psi_{n5} = \frac{1}{2}(w,_{n2} + w,_{n3}) \qquad \psi_{n6} = \frac{1}{2}(w,_{n1} + w,_{n3}) \tag{4.15}$$

   where $w,_n = w,_x \cos\beta + w,_y \sin\beta$ and $\psi_n = \psi_x \cos\beta + \psi_y \sin\beta$.

These equations can be gathered in matrix form:

$$\begin{bmatrix}\psi_{x1} & \psi_{y1} & \psi_{x2} & \dots & \psi_{y6}\end{bmatrix}^T = \mathbf{T}\begin{bmatrix}w_1 & w_{,x1} & w_{,y1} & w_2 & \dots & w_{y3}\end{bmatrix}^T \tag{4.16}$$

where:

$$\mathbf{T} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\frac{3}{2}\frac{s_{12}}{L_{12}} & \frac{c_{12}^2}{2}-\frac{s_{12}^2}{4} & \frac{3}{4}c_{12}s_{12} & -\frac{3}{2}\frac{s_{12}}{L_{12}} & \frac{c_{12}^2}{2}-\frac{s_{12}^2}{4} & \frac{3}{4}c_{12}s_{12} & 0 & 0 & 0 \\
-\frac{3}{2}\frac{c_{12}}{L_{12}} & \frac{3}{4}c_{12}s_{12} & \frac{s_{12}^2}{2}-\frac{c_{12}^2}{4} & \frac{3}{2}\frac{c_{12}}{L_{12}} & \frac{3}{4}c_{12}s_{12} & \frac{s_{12}^2}{2}-\frac{c_{12}^2}{4} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{3}{2}\frac{s_{23}}{L_{23}} & \frac{c_{23}^2}{2}-\frac{s_{23}^2}{4} & \frac{3}{4}c_{23}s_{23} & -\frac{3}{2}\frac{s_{23}}{L_{23}} & \frac{c_{23}^2}{2}-\frac{s_{23}^2}{4} & \frac{3}{4}c_{23}s_{23} \\
0 & 0 & 0 & -\frac{3}{2}\frac{c_{23}}{L_{23}} & \frac{3}{4}c_{23}s_{23} & \frac{s_{23}^2}{2}-\frac{c_{23}^2}{4} & \frac{3}{2}\frac{c_{23}}{L_{23}} & \frac{3}{4}c_{23}s_{23} & \frac{s_{23}^2}{2}-\frac{c_{23}^2}{4} \\
-\frac{3}{2}\frac{s_{31}}{L_{31}} & \frac{c_{31}^2}{2}-\frac{s_{31}^2}{4} & \frac{3}{4}c_{31}s_{31} & 0 & 0 & 0 & \frac{3}{2}\frac{s_{31}}{L_{31}} & \frac{c_{31}^2}{2}-\frac{s_{31}^2}{4} & \frac{3}{4}c_{31}s_{31} \\
\frac{3}{2}\frac{c_{31}}{L_{31}} & \frac{3}{4}c_{31}s_{31} & \frac{s_{31}^2}{2}-\frac{c_{31}^2}{4} & 0 & 0 & 0 & -\frac{3}{2}\frac{c_{31}}{L_{31}} & \frac{3}{4}c_{31}s_{31} & \frac{s_{31}^2}{2}-\frac{c_{31}^2}{4}
\end{bmatrix} \tag{4.17}$$

with $c_{ij} = \cos\beta_{ij}$ and $s_{ij} = \sin\beta_{ij}$.

It follows that the midplane curvatures can be calculated from the rotations only:

$$\begin{bmatrix}w_{,xx}\\ w_{,yy}\\ w_{,xy}\end{bmatrix} = \begin{bmatrix}\psi_{x,x}\\ \psi_{y,y}\\ \frac{\psi_{x,y}+\psi_{y,x}}{2}\end{bmatrix} = \begin{bmatrix}N_{1,x} & 0 & N_{2,x} & 0 & \dots & N_{6,x} & 0\\ 0 & N_{1,y} & 0 & N_{2,y} & \dots & 0 & N_{6,y}\\ \frac{N_{1,y}}{2} & \frac{N_{1,x}}{2} & \frac{N_{2,y}}{2} & \frac{N_{2,x}}{2} & \dots & \frac{N_{6,y}}{2} & \frac{N_{6,x}}{2}\end{bmatrix}\begin{bmatrix}\psi_{x1}\\ \psi_{y1}\\ \psi_{x2}\\ \vdots\\ \psi_{y6}\end{bmatrix} \tag{4.18}$$

$$\begin{bmatrix}w_{,xx}\\ w_{,yy}\\ w_{,xy}\end{bmatrix} = [\partial\mathbf{N}]\begin{bmatrix}\psi_{x1}\\ \psi_{y1}\\ \psi_{x2}\\ \vdots\\ \psi_{y6}\end{bmatrix} \tag{4.19}$$

Since the shape functions are defined in the reference coordinates $\xi$ and $\eta$, the Jacobian matrix is needed to calculate the derivatives as a function of the actual coordinates $x$ and $y$:

$$\begin{bmatrix}N_{i,x}\\ N_{i,y}\end{bmatrix} = \mathbf{J}^{-1}\begin{bmatrix}N_{i,\xi}\\ N_{i,\eta}\end{bmatrix} \tag{4.20}$$

where [**J**] is the Jacobian matrix which is equal to:

$$\mathbf{J} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix} \tag{4.21}$$

where $x_i$ (resp. $y_i$) is the x-coordinate (resp. y-coordinate) of the node $i$ of the triangular element.

Equations 4.16 and 4.19 can be gathered to calculate the midplane curvatures in function of the degrees of freedom:

$$\begin{bmatrix} w_{,xx} \\ w_{,yy} \\ w_{,xy} \end{bmatrix} = [\partial \mathbf{N}]\, \mathbf{T} \begin{bmatrix} w_1 \\ w_{,x1} \\ w_{,y1} \\ w_2 \\ \vdots \\ w_{y3} \end{bmatrix} \tag{4.22}$$

The moments are calculated from the curvatures using Kirchhoff Plate Theory:

$$\begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \mathbf{D} \begin{bmatrix} w_{,xx} \\ w_{,yy} \\ w_{,xy} \end{bmatrix} = \mathbf{D}\, [\partial \mathbf{N}]\, \mathbf{T} \begin{bmatrix} w_1 \\ w_{,x1} \\ w_{,y1} \\ w_2 \\ \vdots \\ w_{y3} \end{bmatrix} \tag{4.23}$$

where **D** is the flexural rigidity matrix for a homogeneous, isotropic material:

$$\mathbf{D} = D \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1 - \nu \end{bmatrix} \tag{4.24}$$

with $D$, the bending stiffness, and $\nu$ the Poisson's ratio.

The element stiffness matrix is equal to:

$$[\mathbf{k_e}] = \int_{S_e} \mathbf{T}^T\, [\partial \mathbf{N}]^T\, \mathbf{D}\, [\partial \mathbf{N}]\, \mathbf{T} dS \tag{4.25}$$

where $S_e$ is the surface of the element. This integral can be calculated using a Gauss quadrature rule. The global stiffness matrix can be assembled from each element stiffness matrix by adding the contributions at each node.

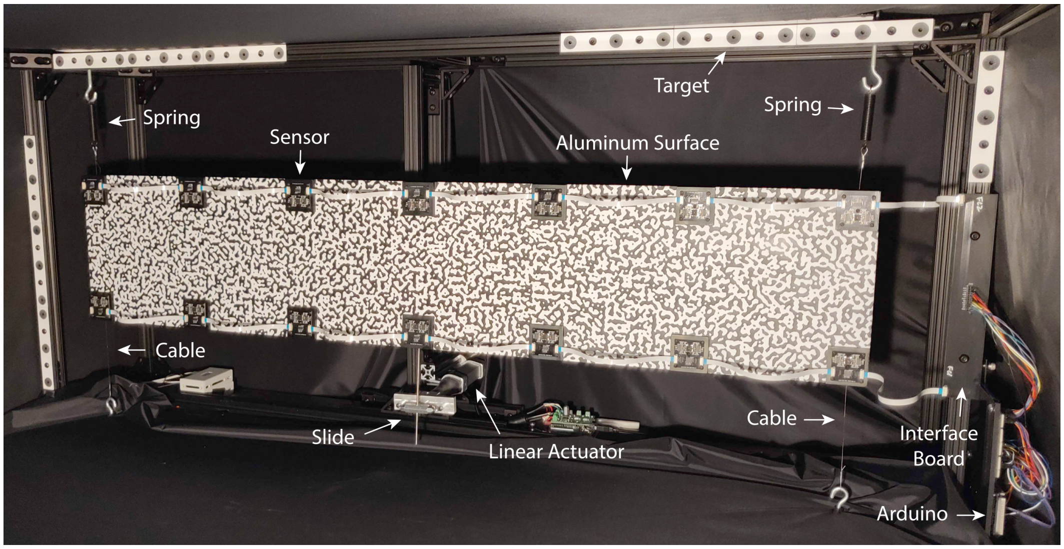


Figure 4.3: Picture of the experiment showing the surface to reconstruct, its boundary conditions and the light sensor system.

## 4.4 Experimental Validation

This new algorithm formulation is tested on the experiment presented in chapter 3 of this thesis. The surface to reconstruct is an aluminum plate (0.016” thick, 50” long and 10” wide) fitted with 14 light sensors (see figure 4.3). A LED light is placed about 2 m in front of the structure to generate the angle measurement from the light sensors.

More details can be found in chapter 3, especially about the calibration of the sensors which measure angles from the light with a precision of 0.5°. They are positioned on rigid aluminum mounts to prevent them from moving during the deformation of the structure. This offset is included in the measurement equations in the algorithm (see chapter 3).

The reconstructed shape from the algorithm is compared to a secondary measurement system which also measures the shape of the structure with a better precision of about an order of magnitude (50 $\mu m$ RMS). This system uses two cameras to image the random pattern printed on the structure. The position of a point is triangulated using Digital Image Correlation (DIC). This comparison estimates the accuracy of the reconstructed shape of the algorithm.

The shape of the structure is controlled by two linear actuators placed in the middle of the structure: one at the center, linked to the surface by a pivot and one below the structure linked by a pivot and slider system. The actuators can impose two types of motion: bending which is characterized by the motion of the top actuator, and torsion which is characterized by the difference of motion between the actuators.

Multiple deformed shapes are imposed by the actuators, combining different bending and torsion modes. For each static shape, 100 measurements were taken by the sensors leading to 100 slightly different reconstructed shapes due to their noise. The RMS error of each shape is calculated to obtain an understanding of the overall error of the reconstruction.

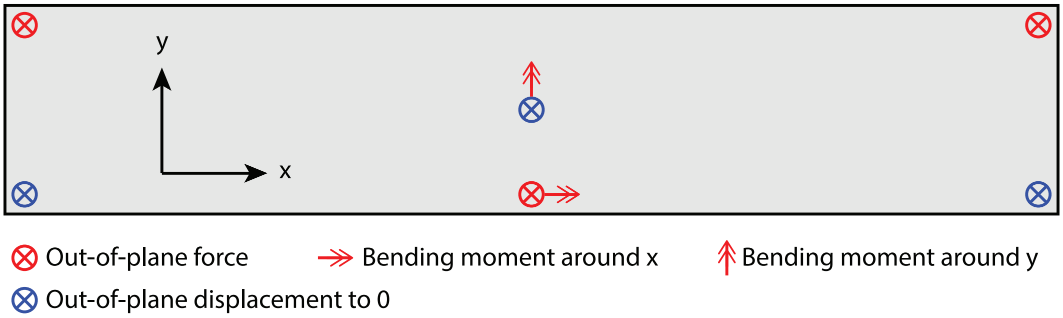


Figure 4.4: Boundary conditions of the FE model. The amplitude of each load is left as an unknown, solved by the angle measurements.

Figure 4.4 shows the different load boundary conditions that are integrated with the new reconstruction algorithm. Their location is used to generate the matrix $\mathbf{F}_d$ (equation 4.4). A total of 5 loads are considered and recreate the boundary conditions applied to the experimental structure. In order to fix the shape to its local reference frame, the out-of-plane deflection of the nodes shown in figure 4.4 in blue are fixed to 0. Note that these constraints are associated with reaction forces that are load boundary conditions (but not explicitly calculated by the model).

The algorithm has a numerical singularity which is associated with the distance from the surface to the light. More details on this singularity can be found in chapter 2. To circumvent this issue, the distance of the central point of the surface (which coincides with the origin of the reference system, see figure 4.1) is measured with the DIC camera system. The translation vector of equation 4.2 is then generated by two rotation matrices:

$$\mathbf{t} = \mathbf{R}_x \mathbf{R}_y \mathbf{t}_0 \tag{4.26}$$

where $\mathbf{t}_0$ is the original position of the center of the structure, $\mathbf{R}_x$ is the rotation matrix around the x-axis, and $\mathbf{R}_y$ is the rotation matrix around the y-axis. The two angles defining the rotation matrices are used in the unknown vector (equation 4.5) instead of the 3 coordinates of the translation vector. Note that in order to match the results of the algorithm to the DIC measurement, the position vector $\mathbf{r}$ (equation 4.2) is rotated by the opposite angles during post-processing:

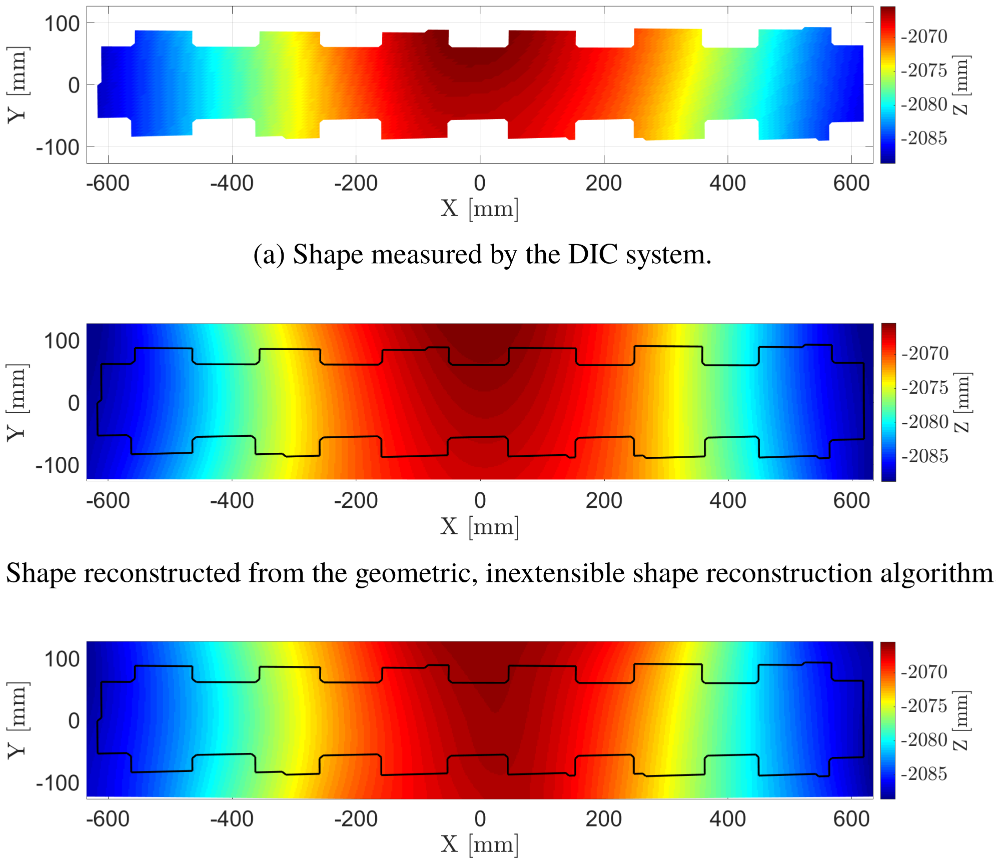
$$\mathbf{r}_{\text{final}} = \mathbf{R}_y^T \mathbf{R}_x^T \mathbf{r} \tag{4.27}$$

The original algorithm performed well for simple deformation such as pure bending or pure torsion. Figure 4.5 shows the results of the experiment for a 20 mm deflection of the structure. The reconstructed shapes of both algorithms (figures 4.5b and 4.5c) are compared to the DIC shape (figure 4.5a). The results are very similar qualitatively. A detailed map of the average error across the surface is shown in figure 4.6. The FE based algorithm does a better job on the left of the structure and lower right which leads to a slightly better accuracy (0.91 mm vs. 1mm). Similar results are shown for pure torsion of the structure in figure 4.7. The results are almost identical with an RMS error of about 0.55 mm for both. The error maps shown in figure 4.8 display relatively similar results. While the FE based algorithm performs better on the left, the geometric, inextensible one seems to be best on the right.

Figures 4.9 and 4.11 show the results for more complex shapes that combine torsion and bending. The error of the reconstruction using the geometric, inextensible shape reconstruction algorithm were dominated by the deformation around the center of the structure. The new, FEM based algorithm does a much better job capturing these local inflections which leads to a smaller RMS errors. This is especially noticeable in figure 4.11 where the yellow line follows a relatively smooth arc for the solution of the geometric, inextensible shape reconstruction algorithm. The solution of the FEM based algorithm creates a yellow line with an inflection point as measured by the DIC.

This point is highlighted by the error maps of the reconstructed shape (figures 4.10 and 4.12). In both cases, the FE based algorithm is more accurate around the center of the structure which leads to a better RMS error overall.

Figure 4.13 shows the average RMS error for all the different shapes reconstructed for this experiment. Overall, the FEM based algorithm provides much better results than the geometry based one with about a factor 1.5 increase in accuracy (about 42 %). The average RMS error for the original algorithm over the different experiments is 1.25 mm and decreases to 0.9 mm with the finite element approach. The accuracy improvements go as high as 70% in the case of the *10 mm / 5.2°* shape. Higher errors still appear for more complex shapes, but the difference is much smaller.



(a) Shape measured by the DIC system.

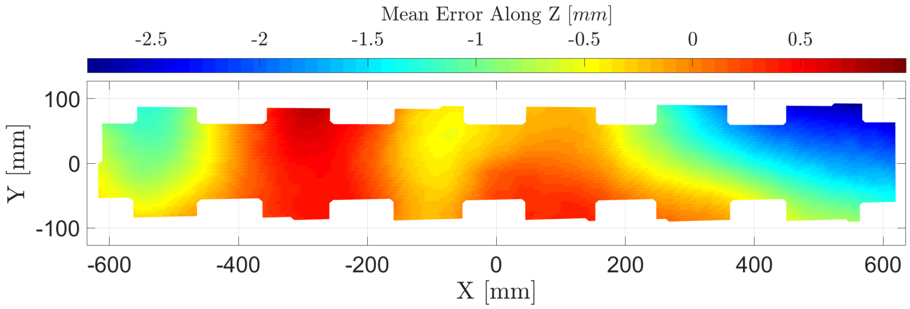
(b) Shape reconstructed from the geometric, inextensible shape reconstruction algorithm.

(c) Shape reconstructed from the finite-element based shape reconstruction algorithm.

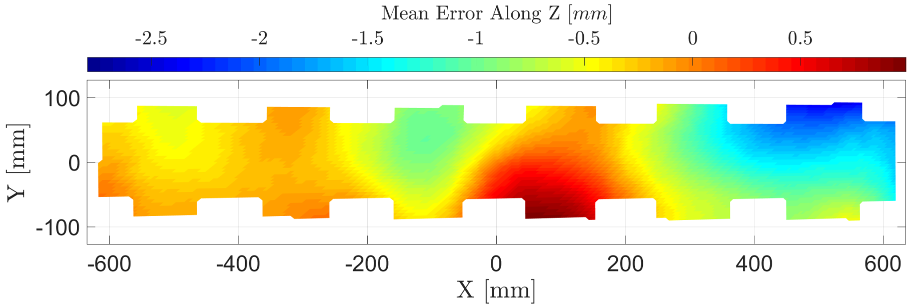
Figure 4.5: Measured and reconstructed shapes for a 20 mm deflection of the structure.

### Computational time

The geometric, inextensible shape reconstruction algorithm takes about 150 ms to calculate the position of the control points. The FE based algorithm takes about 30 s to estimate all the entries of the vector of unknowns. The difference in computational time is mostly due to the number of unknowns (45 vs. 1690) and that a finite difference scheme was used to compute most of the Jacobian matrices. More work could be done to optimize the speed of the algorithm for applications in which timing is crucial. Simpler and faster methods than the Levenberg-Marquardt one, explicit calculation of the Jacobian matrix, or linearizations of the constraints are some of the possible ways to increase the speed of the presented shape reconstruction algorithms.



(a) Mean error of shapes reconstructed from the geometric, inextensible shape reconstruction algorithm.



(b) Mean error of shapes reconstructed from the finite-element based shape reconstruction algorithm.

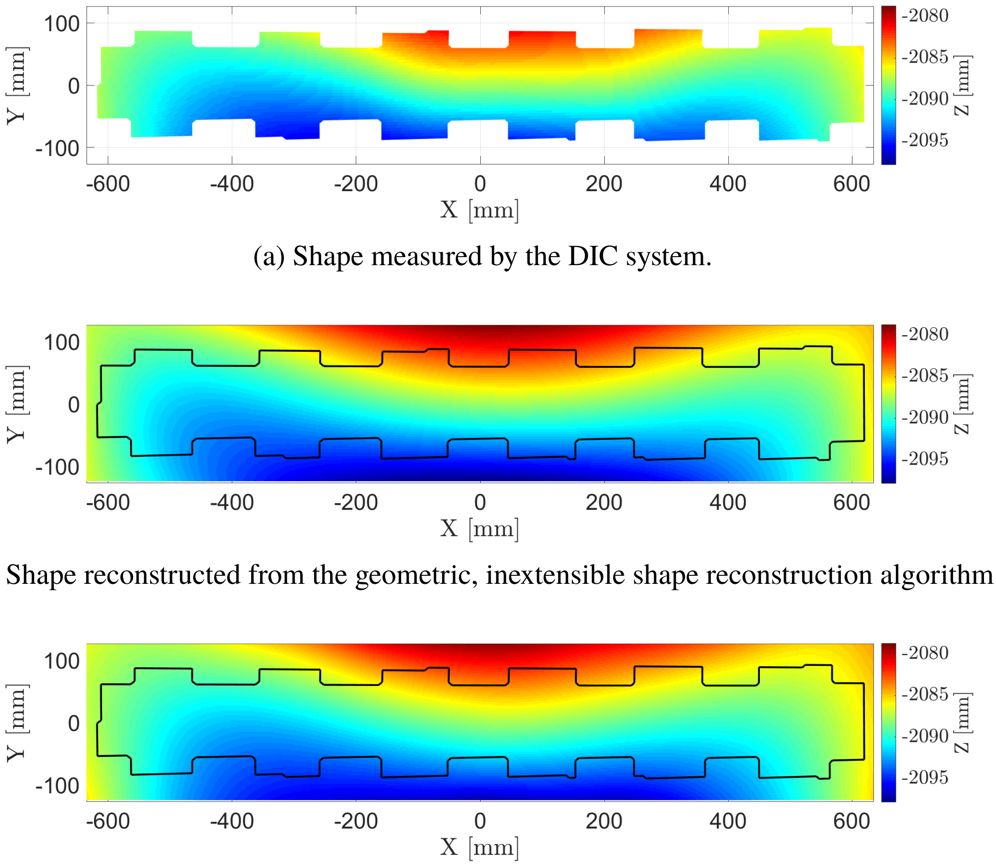
Figure 4.6: Error of reconstructed shapes for a 20 mm deflection of the structure.

## 4.5 Conclusion and Discussions

This chapter presented an updated version of the shape reconstruction algorithm introduced in chapter 2. Experimental results using this original algorithm in chapter 3 showed that local shape deformations due to the boundary conditions on the structure cannot be accurately solved for. This consequently limited the accuracy of the results.

By introducing a structural mechanics model of the structure, these local deformations can be predicted. The equilibrium equations effectively replace the intextensibility conditions previously introduced to remove the singularities of the measurement equations. The framework of the algorithm remains mostly unchanged and only these inextensibility equations are replaced with the static equilibrium derived from a finite-element model.

The new algorithm is used on the raw measurement data from the experiment detailed in chapter 3 and the reconstructed shapes are compared to the results shown



(a) Shape measured by the DIC system.

(b) Shape reconstructed from the geometric, inextensible shape reconstruction algorithm.

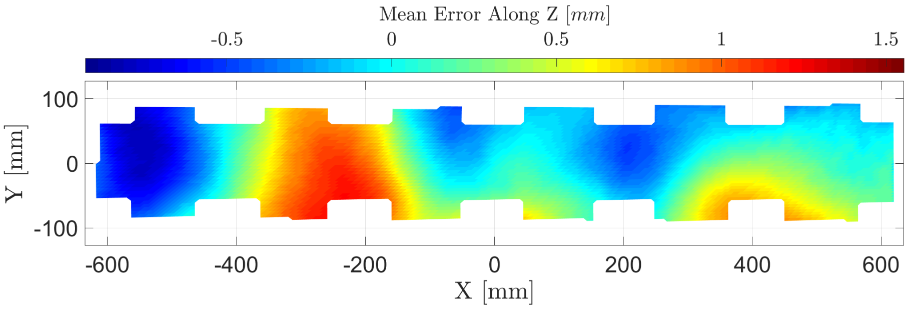
(c) Shape reconstructed from the finite-element based shape reconstruction algorithm.

Figure 4.7: Measured and reconstructed shapes for a “pure” 5.2° torsion of the structure.

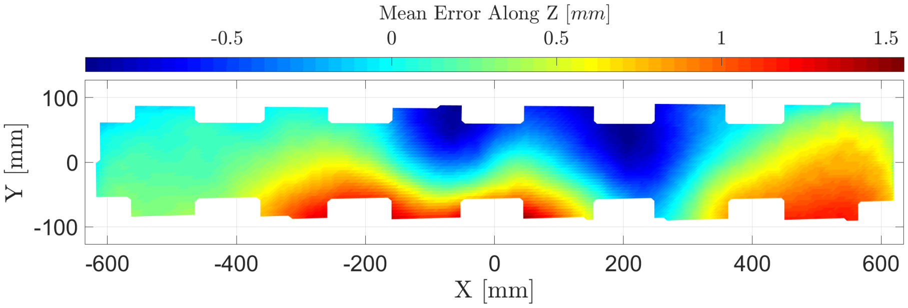
in the paper. A net improvement of accuracy by over 40% was shown. The local shape deformations are better estimated and while the error previously increased when the deformed shape became more complex, this sensitivity has been reduced.

The finite element model is however restricted to small deformations. More work needs to be done to make this algorithm more general and include a geometrically nonlinear finite element model instead. This would add complexity to the system of equations and make the algorithm potentially slower, but more suited to a wider range of applications. The performance of the algorithm should then be tested to more complex geometries to better understand its limits.

The system of equations generating the algorithm can be linearized for small deformations and if the incident light on the structure is collimated (the direction of the light is identical across the structure). This linear system would greatly simplify and



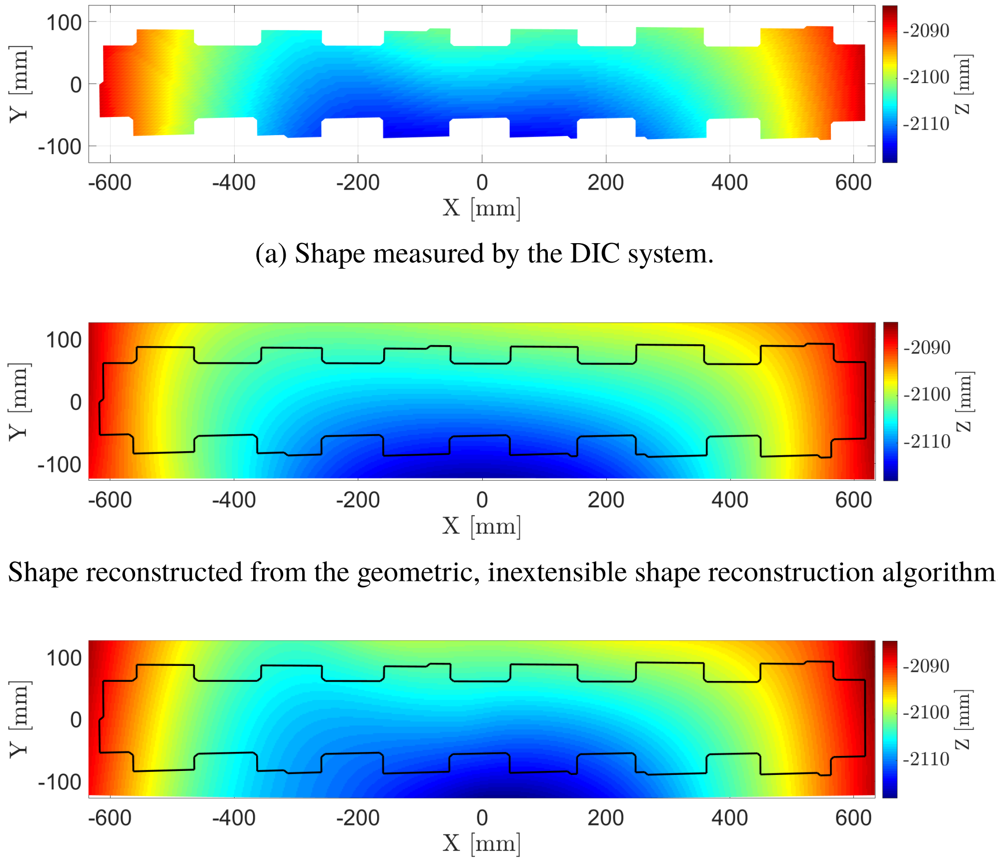
(a) Mean error of shapes reconstructed from the geometric, inextensible shape reconstruction algorithm.



(b) Mean error of shapes reconstructed from the finite-element based shape reconstruction algorithm.

Figure 4.8: Error of reconstructed shapes for a “pure” 5.2° torsion of the structure.

accelerate the solving process which could be advantageous for space applications where computational resources are limited.

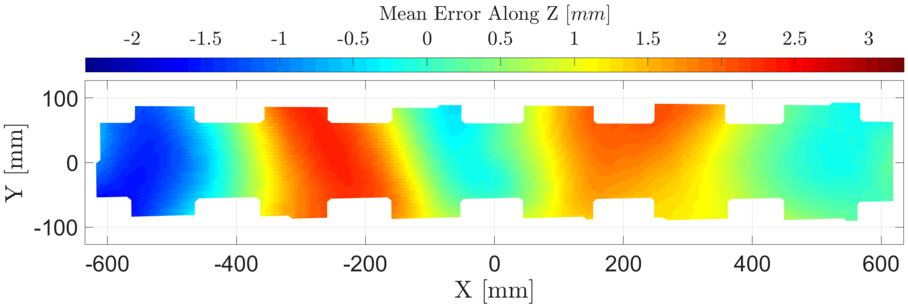


(a) Shape measured by the DIC system.

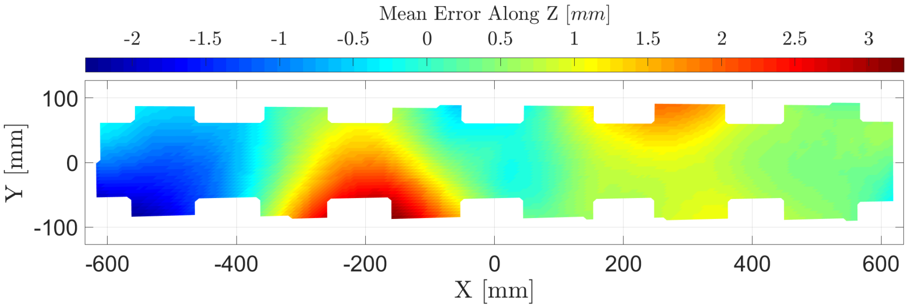
(b) Shape reconstructed from the geometric, inextensible shape reconstruction algorithm.

(c) Shape reconstructed from the finite-element based shape reconstruction algorithm.

Figure 4.9: Measured and reconstructed shapes for a -20 mm deflection and 5.2° torsion of the structure.

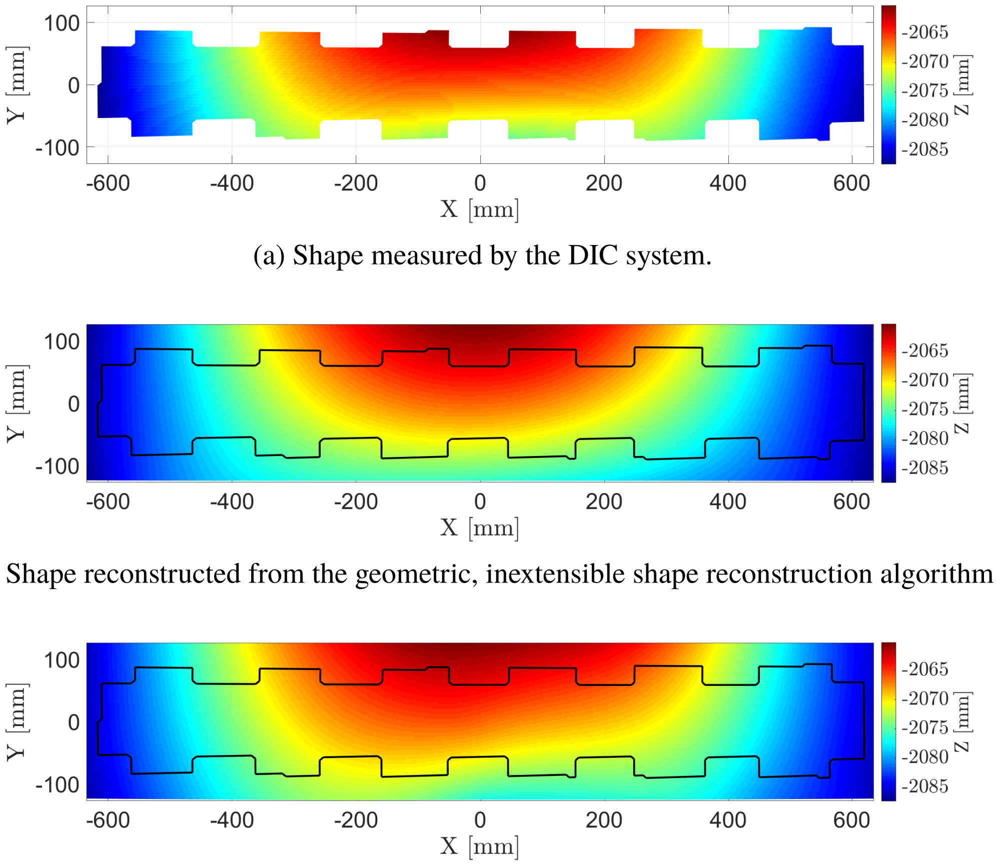


(a) Mean error of shapes reconstructed from the geometric, inextensible shape reconstruction algorithm.



(b) Mean error of shapes reconstructed from the finite-element based shape reconstruction algorithm.

Figure 4.10: Error of reconstructed shapes for a -20 mm deflection and 5.2° torsion of the structure.

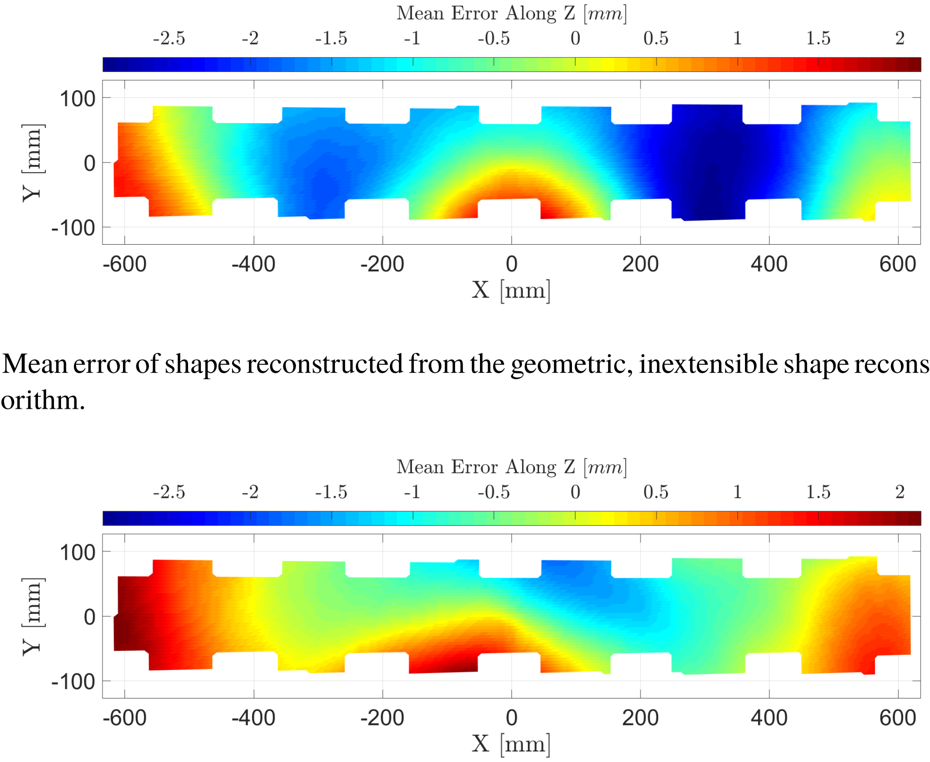


(a) Shape measured by the DIC system.

(b) Shape reconstructed from the geometric, inextensible shape reconstruction algorithm.

(c) Shape reconstructed from the finite-element based shape reconstruction algorithm.

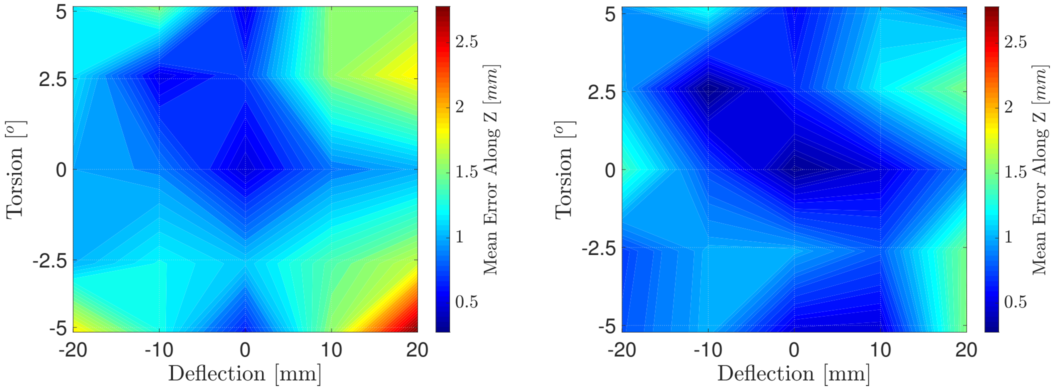
Figure 4.11: Measured and reconstructed shapes for a 20 mm deflection and 5.2° torsion of the structure.



(a) Mean error of shapes reconstructed from the geometric, inextensible shape reconstruction algorithm.

(b) Mean error of shapes reconstructed from the finite-element based shape reconstruction algorithm.

Figure 4.12: Error of reconstructed shapes for a 20 mm deflection and 5.2° torsion of the structure.



(a) Mean RMS error using the geometric, inextensible shape reconstruction algorithm.

(b) Mean RMS error using the finite-element based shape reconstruction algorithm.

Figure 4.13: Mean RMS error of the reconstructed shape for different applied deflections and torsions.

*Chapter 5*

# DYNAMIC SHAPE MEASUREMENT OF A LARGE SPACECRAFT USING A KALMAN FILTER

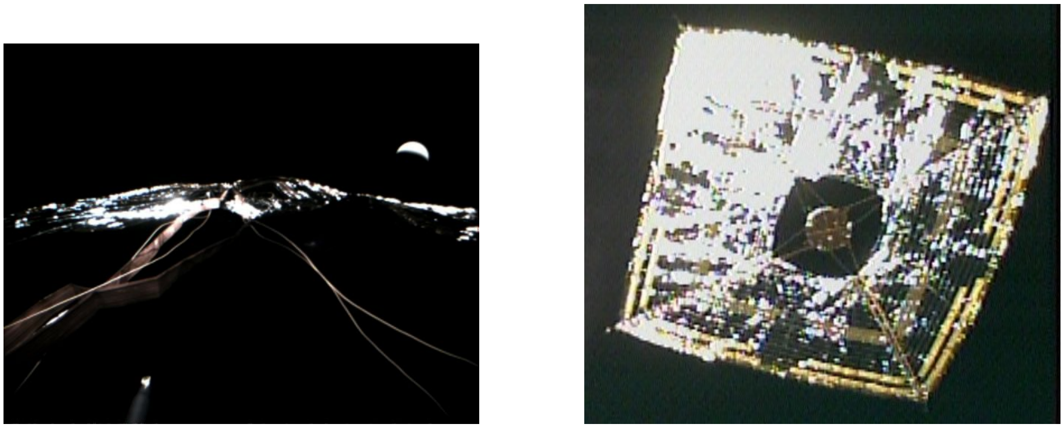
## 5.1 Introduction

The size of spacecrafts is limited by the size of the rocket placing them into orbit. Large structures in space involve either the assembly or the deployment of parts which is often complicated, heavy or expensive. Relatively simple and conservative designs often using joints are usually considered. Large structures are key of the future of space as they open a large field of applications [9, 43].

Many applications are driven by the shape accuracy of the structure. This is usually achieved by designing stiff structures [37] which tend to make the satellite heavy (primary apperture of JWST), but more aggressive designs are being investigated (solar sails such as IKAROS (JAXA), NanoSail (NASA), or LightSail (Planetary Society)). Such gossamer structures are very light and highly packageable which is advantageous when launching them into space.

New concepts are trying to use gossamer structures while still requiring high shape accuracy. To do so, some enforce the flatness of the structure by tensioning it [90]. This however requires tens of newtons of force which eventually have to be carried as compression in some structural elements, which tend to be bulky and heavy to avoid buckling. Vibration suppression is another method being studied to maintain a stable shape [70]. Bending stiff elements can be included in the design to limit the deformation of the structure. The roll-out solar array experiment (ROSA) demonstrated a new design for lightweight solar panels [4]. The Space Solar Power Project at Caltech is developing a deployable spacecraft structure that, once deployed to its planar configuration, can generate DC power from the sunlight and beam it back to Earth using an RF antenna array [3, 30].

Measuring the shape of these lightweight deformable structures can greatly improve their performance and may enable their use in some applications. For example, the shape of a solar sail directly correlates with the direction and amplitude of its acceleration. This measure becomes useful for guidance and navigation controllers especially for long duration missions [10]. Large antenna arrays such as those envisaged in the Space Solar Power Project at Caltech do not require a rigid support



(a) Picture from the body-mounted camera of IKAROS.

(b) Picture from the detachable camera of IKAROS.

Figure 5.1: Pictures taken by the cameras on board IKAROS. Credit: JAXA.

structure, provided a phase shift is introduced to correct for the relative displacement of the antennas. This phase correction needs an accurate measurement of the shape of the structure [35].

Cameras have usually been used to measure the shape of large structures in space. IKAROS, for instance, had both body-mounted and detachable cameras (figure 5.1b. The images taken by the cameras were used to retrieve the shape of the sail a posteriori [67, 74, 89]. The vibrational shape and frequency of the ISS solar panels was also measured in space in an experiment aboard the space shuttle using cameras tracking the location of targets [75]. Recently, new concepts have been studied to measure the shape of large structures using sun sensors [10, 84]. These systems are not limited by the small incidence angle of on-board cameras (figure 5.1a) or the short time during which detachable cameras can provide images (figure 5.1b).

We wish to reconstruct the shape of a large planar spacecraft orbiting around the Earth and maneuvering to stay relatively aligned to the Sun. This type of orbit is applicable to solar sails or space solar power satellites [29, 53]. Sun sensors are distributed on the structure and will be used to reconstruct its shape.

Both large rotations and vibrations are to be measured. A Kalman filter is used to estimate the shape of the structure by combining the measurements with a dynamic, mechanical model.

The first section defines the problem that will be studied in this chapter. The study is limited to a simple spacecraft modeled as a thin plate. The approach is adequate to understand the performance and limitations of the measurement system rather than to consider the detailed performance of a specific space structure.

Section 5.3 will detail the dynamic model of the structure. The out-of-plane deformation and vibration of the spacecraft are captured by a linear plate bending finite element model. Because the spacecraft can undergo large rotations, this model includes such rigid-body effects [76, 77]. Initially non-linear, a linear dynamic model of the spacecraft can be achieved from simple assumptions that will be detailed. The number of degrees of freedom of the model is largely reduced using a Craig-Bampton model reduction which becomes computationally useful for the implementation of a Kalman filter [32, 68].

The equations of the Kalman filter are detailed in section 5.4. The finite element model is integrated to generate a discrete time state model used in the filter [14, 20]. A state vector represents the degrees of freedom of the model that are computed through this filter and updated by the measurements. The measurement equations defining the relation between the state vector and the angles measured by the sun sensors are also explained.

Finally, the performance of such a system is analyzed for a model of the Space Solar Power Satellite in section 5.5 . While the architecture of this satellite is complex [30, 31, 45], its dynamics are simplified according to the model detailed in section 5.3. Simulations are done on a $25 \times 25$ m version of the spacecraft. The accuracy of such a system and the gain in performance compared to a rigid-body model is determined, and the influence of the number of sensors is shown. Studies have also been conducted to show the robustness of the filter to possible errors in the modeled stiffness matrix. The effects of errors in the bending stiffness parameters on the accuracy of the system are computed. Finally, it is shown that adding force sensors at the thruster locations would greatly improve the accuracy of the system.

## 5.2 Definition of the Problem

We assume that the spacecraft is square with its mass uniformly distributed except a possible point mass at its center (this represents a central hub). Its bending stiffness is also assumed homogeneous and isotropic such that the mechanical behavior of the planar spacecraft resembles a plate in bending. A simple schematic of such a spacecraft is shown in figure 5.2.

The spacecraft can rotate off the Sun axis either for guidance, station keeping or to optimize its output (in the case of a space solar power satellite). The Sun is assumed to remain within the field of view of the sensors. For simplicity, we assume that

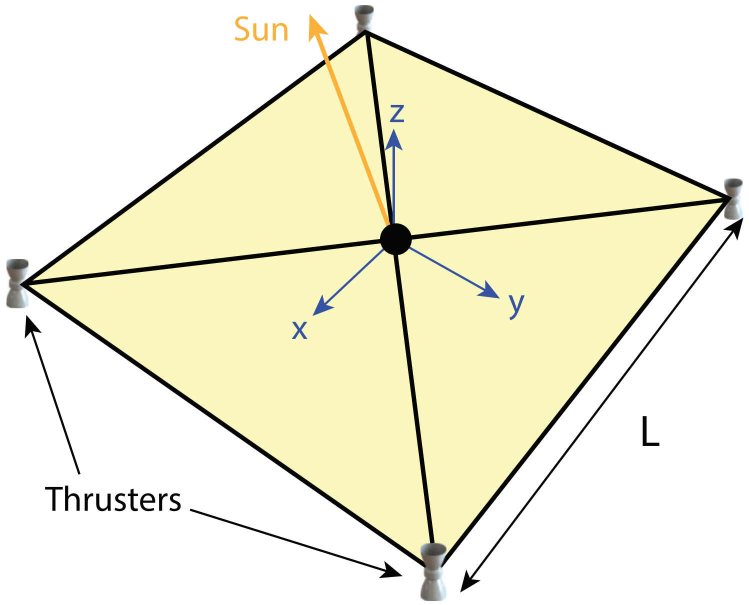


Figure 5.2: Model spacecraft. The square planar structure is homogeneous in terms of mass and stiffness, except possibly a point mass at its center. The axes of the structure are defined in the middle and pointing along its edges. Thrusters at each corner provide the necessary actuation for guidance purposes.

these large rotations only happen around the $x$-axis and that the rotations around the other axes are small.

The attitude of the spacecraft is controlled by thrusters placed at each corner of the structure. They are considered massless. When firing, they create a moment that either accelerates the rotation of the spacecraft or decelerates it. Additionally to the rigid rotation, these thrusters will force vibration of the structure whose amplitude can be large enough to affect the performance of the satellite. Note that other type of actuators could be considered such as Reflectivity Control Device that were used on IKAROS to control the solar sail [29].

The large rigid-body rotation and the vibration of the spacecraft are the only phenomenon included in this study. Other effects, not considered in the rest of this chapter, can deform such lightweight, deployable structure: jamming of components during the deployment, thermal loads, or the solar pressure which is often non-negligible for such large structures [37].

In order to reconstruct the vibrating shape of the spacecraft, we will use a Kalman filter which merges the measurements and a dynamic model of the structure.

## 5.3 Finite Element Model of the Spacecraft

A model of the spacecraft is needed by the Kalman filter that will be presented in section 5.4, in order to predict the behavior of the system between measurements. These predictions are then corrected at every measurement interval. The shape of the spacecraft is dictated by a large rigid rotation around its center of mass and off-plane vibrations, both generated by external loads such as thruster forces. We assume that the structure was designed to stay relatively planar, i.e. any out-of-plane deflection is assumed small.

Both the vibration of the structure and its large rotation are captured in a single model. A finite displacement model can capture both effects by assuming large displacements of the nodes [7]. This formulation is, however, geometrically non-linear and can be complicated especially when off-plane vibrations can be assumed small and accurately modeled by a linear, small displacement model. An extended model can be formulated by integrating a finite element model with large rigid-body motions and will be presented in the next sub-section. While non-linear, this combined model can be linearized under certain assumptions that will be presented.

This linearized model is further reduced to decrease its number of degrees of freedom. Only low frequency modes are sufficient to capture the vibrating shape of the structure. This reduced order model is very advantageous in a Kalman filter formulation as it greatly accelerates numerical calculations.

### Floating Frame of Reference Formulation

In this sub-section, we detail the formulation that combines a local finite-element model with large rigid-body motions. It has been originally developed for multi-body systems in [76] and is called the Finite Element Floating Frame of Reference Formulation. The details of the derivation can be found in [76], and only the main definitions and results are presented here.

Let $(X, Y, Z)$ be an inertial reference frame and $(x, y, z)$ be the local, non-inertial reference frame of the spacecraft as shown in figure 5.3. The position of a point P on the structure $\mathbf{r}^P$ is defined in the inertial reference frame $(X, Y, Z)$ by:

$$\mathbf{r}^P = \mathbf{A}\mathbf{u}^P + \mathbf{t} \tag{5.1}$$

where $\mathbf{A}$ is the rigid-body rotation matrix, $\mathbf{u}^P$ is the position of point P in the local reference frame $(x, y, z)$, and $\mathbf{t}$ is the rigid-translation of the structure expressed in $(X, Y, Z)$.

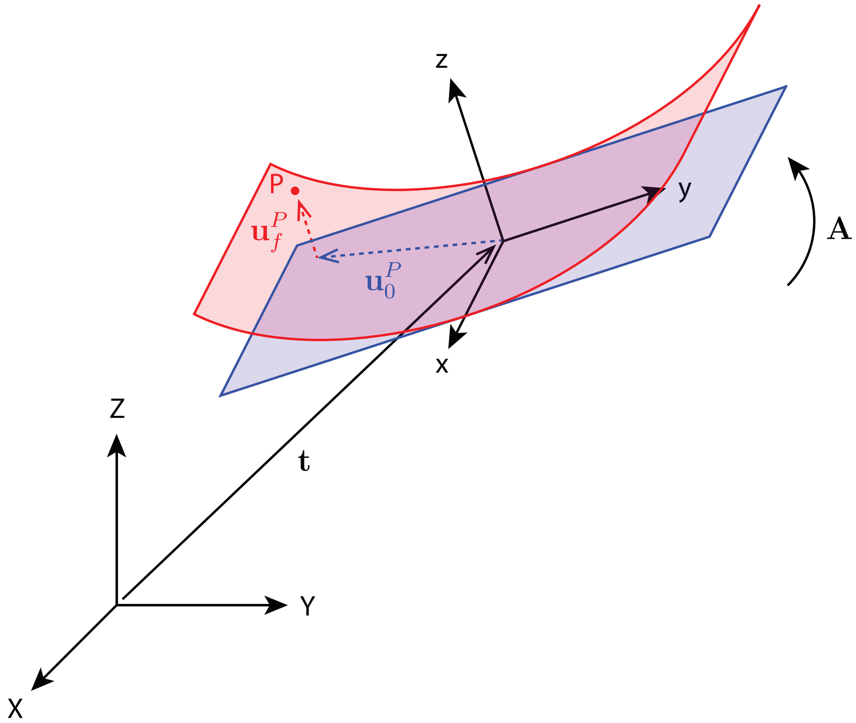


Figure 5.3: Description of the Floating Frame of Reference Formulation (FFR). The deformation of the structure is defined locally in $(x, y, z)$. It is then moved rigidly in the inertial frame $(X, Y, Z)$.

Small vibrations of the structure are defined by a finite element model in the local reference frame. Any rigid body mode of this model has to be suppressed to avoid redundancies with the translation vector and rotation matrix of equation 5.1.

The position of a point on the structure in the local reference frame is parametrized by the shape functions of the finite element model:

$$\mathbf{u}^P = \mathbf{u}_0^P + \mathbf{u}_f^P = \mathbf{u}_0^P + \mathbf{N}^P \mathbf{q}_f \tag{5.2}$$

where $\mathbf{u}_0^P$ is the initial position of point P in the undeformed state, $\mathbf{N}^P$ is the matrix of space-dependent shape functions used in the local finite element model evaluated at point P and $\mathbf{q}_f$ is the time-dependent vector of degrees of freedom of the finite element model.

The rotation matrix $\mathbf{A}$ can be defined according to different conventions: Euler angles, Rodrigues' parameters, Euler parameters, quaternions, etc [41]. Let $\mathbf{q}_\theta$ be the vector of the chosen parameters. The rotation matrix is explicitly defined by these parameters: the $x - y - z$ Tait–Bryan angles sequence generates a rotation matrix as follows:

$$\mathbf{A} = \mathbf{A}_z\left(\theta_z\right)\mathbf{A}_y\left(\theta_y\right)\mathbf{A}_x\left(\theta_x\right) = \begin{bmatrix} c_x c_z & -c_y s_z & s_y \\ c_z s_x s_y + c_x s_z & c_x c_z - s_x s_y s_z & -c_y s_x \\ -c_x c_z s_y + s_x s_z & c_z s_x + c_x s_y s_z & c_x c_y \end{bmatrix} \tag{5.3}$$

where $\mathbf{A}_x$ (resp. $\mathbf{A}_y$, $\mathbf{A}_z$) is the basic rotation matrix around the $x$-axis (resp. $y$, $z$-axis), $c_{x,y,z} = \cos\theta_{x,y,z}$, $s_{x,y,z} = \sin\theta_{x,y,z}$. The angles $\theta_x$, $\theta_y$ and $\theta_z$ are commonly known as the roll, pitch and yaw angles. The rotation parameters can be gathered in vector form: $\mathbf{q}_\theta = [\theta_x \quad \theta_y \quad \theta_z]^T$.

The coupling between the rigid motion and the local deformation of the structure is fully captured by the mass matrix and a quadratic velocity force vector. To estimate the mass matrix, the expression for velocity vector at point P needs to be written. It can be written by taking the time derivative of equation 5.1:

$$\dot{\mathbf{r}}_P = \dot{\mathbf{A}}\mathbf{u}^P + \mathbf{A}\mathbf{N}^P\dot{\mathbf{q}}_f + \dot{\mathbf{t}} = \begin{bmatrix} \mathbf{I}_{3\times3} & -\mathbf{A}[\mathbf{u}^P]_\times\mathbf{G} & \mathbf{A}\mathbf{N}^P \end{bmatrix} \begin{bmatrix} \dot{\mathbf{t}} \\ \dot{\mathbf{q}}_\theta \\ \dot{\mathbf{q}}_f \end{bmatrix} \tag{5.4}$$

where $\mathbf{I}_{3\times3}$ is the $3\times3$ identity matrix, $[\mathbf{u}^P]_\times$ is the skew-symmetric tensor of vector $\mathbf{u}^P$, and $\mathbf{G}$ is the linear operator that transforms the time derivative of the rotation parameters $\dot{\mathbf{q}}_\theta$ to the angular velocity vector $\boldsymbol{\omega} = \mathbf{G}\,\dot{\mathbf{q}}_\theta$.

The linear operator $\mathbf{G}$ depends on the parametric representation of the rotation $\mathbf{q}_\theta$. For the $x - y - z$ Tait–Bryan angles sequence, one can show that it is equal to:

$$\mathbf{G} = \begin{bmatrix} c_y c_z & -s_z & 0 \\ c_y s_z & c_z & 0 \\ -s_y & 0 & 1 \end{bmatrix} \tag{5.5}$$

**Equations of motion**

The general form of the equations of motion of the spacecraft can be written as:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F}_a + \mathbf{F}_v \tag{5.6}$$

where $\mathbf{q} = [\mathbf{t}^T \quad \mathbf{q}_\theta^T \quad \mathbf{q}_f^T]^T$ is the vector of degrees of freedom of the model, $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ is the stiffness matrix, $\mathbf{F}_a$ is the vector of applied forces, and $\mathbf{F}_v$ is the quadratic velocity vector which represents the Coriolis and centrifugal forces.

### Mass Matrix

The mass matrix **M** can be calculated by writing the expression for the kinetic energy in the inertial reference frame. This leads to a highly non-linear expression:

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_{tt} & \mathbf{m}_{t\theta} & \mathbf{m}_{tf} \\ \mathbf{m}_{\theta t} & \mathbf{m}_{\theta\theta} & \mathbf{m}_{\theta f} \\ \mathbf{m}_{ft} & \mathbf{m}_{f\theta} & \mathbf{m}_{ff} \end{bmatrix} \tag{5.7}$$

where:

- $\mathbf{m}_{tt}$ is the mass of the spacecraft:

  $$\mathbf{m}_{tt} = m\,\mathbf{I}_{3\times 3} \tag{5.8}$$

- $\mathbf{m}_{\theta t}$ is the first moment of mass and vice versa, $\mathbf{m}_{t\theta} = \mathbf{m}_{\theta t}^T$ is the acceleration created by the angular acceleration of the local reference frame:

  $$\mathbf{m}_{t\theta} = -\int_V \rho \mathbf{A}[\mathbf{u}^P]_\times \mathbf{G} dV \tag{5.9}$$

  note that this term is 0 if the origin of the local reference system coincides with the center of mass. This is usually not the case as the deformation of the plate moves the center of mass.

- $\mathbf{m}_{tf} = \mathbf{m}_{ft}^T$ represents the mass associated with each degree of freedom of the finite element model:

  $$\mathbf{m}_{tf} = \int_V \rho \mathbf{A} \mathbf{N}^P dV \tag{5.10}$$

- $\mathbf{m}_{\theta\theta}$ is the moment of inertia (expressed in function of the rotational parameters accelerations):

  $$\mathbf{m}_{\theta\theta} = \int_V \rho \mathbf{G}^T [\mathbf{u}^P]_\times^T [\mathbf{u}^P]_\times \mathbf{G} dV \tag{5.11}$$

- $\mathbf{m}_{\theta f} = \mathbf{m}_{f\theta}^T$ represents the torque generated by the acceleration of the d.o.f. of the finite element model ($\mathbf{m}_{f\theta}$ represents the Euler force):

  $$\mathbf{m}_{\theta f} = -\int_V \rho \mathbf{G}^T [\mathbf{u}^P]_\times^T \mathbf{N}^P dV \tag{5.12}$$

- $\mathbf{m}_{ff}$ is the classical mass matrix of the finite element model:

  $$\mathbf{m}_{ff} = \left(\mathbf{N}^P\right)^T \mathbf{N}^P \tag{5.13}$$

### Stiffness Matrix

Rigid motions do not induce any strains, hence do not contribute to the stiffness matrix. Its expression is directly derived from the finite element model of the structure:

$$\mathbf{K} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{ff} \end{bmatrix} \tag{5.14}$$

where $\mathbf{K}_{ff}$ is the classical stiffness matrix of the finite element model:

$$\mathbf{K}_{ff} = \int_V \left([\partial]\mathbf{N}^P\right)^T \mathbf{D}[\partial]\mathbf{N}^P dV \tag{5.15}$$

with $[\partial]$, the differential operator that calculates the strains from the displacements, and $\mathbf{D}$, the constitutive matrix of elastic constants which, in a case of a homogeneous, isotropic plate model, is the flexural rigidity matrix:

$$\mathbf{D} = D \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{bmatrix} \tag{5.16}$$

### Applied Forces

Let $\mathbf{f}^P$ be a force acting at point P, defined in the local reference frame $(x, y, z)$. $\mathbf{f}^P$ is, for example, the force generated by a thruster attached to the spacecraft. The applied force vector can be expressed by 3 sub-vectors, each associated with the 3 sets of degrees of freedom:

$$\mathbf{F}_a = \begin{bmatrix} \mathbf{F}_{ta} \\ \mathbf{F}_{\theta a} \\ \mathbf{F}_{fa} \end{bmatrix} = \begin{bmatrix} \mathbf{A}\mathbf{f}^P \\ -\mathbf{G}^T[\mathbf{u}^P]_\times^T\mathbf{f}^P \\ \left(\mathbf{N}^P\right)^T \mathbf{f}^P \end{bmatrix} \tag{5.17}$$

where $\mathbf{F}_{ta} = \mathbf{A}\mathbf{f}^P$ is the rigid-body force component in the inertial reference frame, $\mathbf{F}_{\theta a} = -\mathbf{G}^T[\mathbf{u}^P]_\times^T\mathbf{f}^P$ represents the rigid-body moment of the force, and $\mathbf{F}_{fa} = \left(\mathbf{N}^P\right)^T \mathbf{f}^P$ is the discretization of the force onto the nodal coordinates of the finite element model. A similar expression can be written for discrete applied moments. Surface or body forces can also be introduced through integrals over the boundary surface or interior volume of the structure. Finally, if multiple forces are applied, the total force vector is the sum of each individual contribution.

### Quadratic Velocity Vector

The quadratic velocity vector corresponds to the Coriolis and centrifugal forces generated by the rotation of the structure in the inertial reference frame. They create body forces equal to:

$$\mathbf{f}_v = -[\omega]_\times^2 \mathbf{u}^P - 2[\omega]_\times \mathbf{N}^P \dot{\mathbf{q}}_f \tag{5.18}$$

where the first term corresponds to the centrifugal force and the second to the Coriolis force. Note that the Euler force is included in the mass matrix, equation 5.12. The quadratic velocity vector is simply expressed as:

$$\mathbf{F}_v = \begin{bmatrix} \mathbf{F}_{tv} \\ \mathbf{F}_{\theta v} \\ \mathbf{F}_{fv} \end{bmatrix} = \int_V \rho \begin{bmatrix} \mathbf{A}\mathbf{f}_v \\ -\mathbf{G}^T[\mathbf{u}^P]_\times \mathbf{f}_v \\ \left(\mathbf{N}^P\right)^T \mathbf{f}_v \end{bmatrix} dV \tag{5.19}$$

### Assumptions and Linearization of the Equations of Motion

The following assumptions are made to simplify the equations of motion:

1. The out-of-plane displacements of the structure are small. As a result, only the out-of-plane component varies, while the $x, y$-displacement components of a point remain approximately zero (see figure 5.3). The position of a point P in the local reference system (or floating frame of reference) is:

   $$\mathbf{u}^P = \mathbf{u}_0^P + \mathbf{N}^P \mathbf{q}_f = \begin{bmatrix} x_p \\ y_P \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \tilde{\mathbf{N}}^P \end{bmatrix} \mathbf{q}_f \tag{5.20}$$

   The third coordinate $z_P = \tilde{\mathbf{N}}^P \mathbf{q}_f$ is considered small compared to the dimensions of the structure, hence second-order terms in $z_P$ are neglected.

2. The applied forces are assumed to be only along the $z$-axis (see figure 5.3). They are symmetric around the $x$-axis such that they only create large rotations around $x$, small rotations around $y$ (included into the local finite-element model), and no rotation around $z$. Without loss of generality, the angle around $z$ is set to zero.

3. Due to the symmetry of the actuation and the structure, the translation of the center of mass is negligible. This leads to $\mathbf{t} = \mathbf{0}$. Note that this assumption

is equivalent to placing a relatively heavy point mass at the center of mass of the structure (This physically would corresponds to a hub at the center of the structure which is common for this type of satellite). This mass contributes enough to neglect the acceleration of the center of mass of the structure.

4. The angular velocities of the spacecraft are small (slow maneuvers). The size of the spacecraft is relatively large so that the frequency and amplitude of the first modes of vibrations are small (compared to its size) which leads to small out-of-plane velocities of the structure. Both assumptions are used to neglect the quadratic velocity vector where all the terms are second order: $\mathbf{F}_v = \mathbf{0}$.

One can show that, with the aforementioned assumptions, the vector of degrees of freedom is $\mathbf{q} = [\theta_x \quad \mathbf{q}_f^T]^T$. The mass matrix reduces to:

$$\mathbf{M} = \mu \begin{bmatrix} \frac{L^4}{12} & \int_S y \tilde{\mathbf{N}}^P dS \\ \int_S y \left(\tilde{\mathbf{N}}^P\right)^T dS & \int_S \left(\tilde{\mathbf{N}}^P\right)^T \tilde{\mathbf{N}}^P dS \end{bmatrix} \tag{5.21}$$

where $\mu$ is the area density (assumed constant across the spacecraft), and $L$ is the width and length of the structure. Note that the off-diagonal terms correspond to the moments generated by the acceleration of the degrees of freedom of the finite element model (Euler forces). The first diagonal term is the moment of inertia of a rigid structure.

The stiffness matrix is simply derived from the finite element model and is linear, assuming small displacements:

$$\mathbf{K} = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{ff} \end{bmatrix} \tag{5.22}$$

The force vector can be approximated by:

$$\mathbf{F}_a = \sum_F \begin{bmatrix} y^F f^F \\ \left(\mathbf{N}^F\right)^T f^F \end{bmatrix} \tag{5.23}$$

where the index $^F$ represents one of the applied forces at $(x_F, y_F)$. The first entry is the moment of the forces around the center of mass which coincides with the origin of the reference systems.

The linear equation of motion is therefore:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F}_a \tag{5.24}$$

Note that the only difference with a classical small deformation finite-element model is the coupling between the rigid-body angular acceleration and the vibration of the structure (and vice versa).

### Local Finite-Element Model

The Floating Frame of Reference Formulation introduced in the previous sub-section merges a local finite element model with a large rigid-body motion. This finite element model characterizes the small deformation of the spacecraft.

A linear plate-bending finite element model is designed to model the out-of-plane vibrations of the spacecraft. This coincides with the assumptions made in the previous sub-section to linearize the equations of motion.

The deformation of the mid-plane of the plate structure is considered. The mid-surface corresponds to the neutral surface where the in-plane strain components remain zero throughout the deformation. The underformed configuration of the plate is flat and lies in the $xy$-plane. Deformations only occur along the $z$-axis. We further assume that the structure behaves as a homogeneous, isotropic Kirchoff plate (no shear deformation across the thickness). The moment-curvature relationship is therefore of the type:

$$\begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = D \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{bmatrix} \begin{bmatrix} -w,_{xx} \\ -w,_{yy} \\ -w,_{xy} \end{bmatrix} \tag{5.25}$$

where $D$ is the bending stiffness and $\nu$ is the Poisson's ratio.

Each node of the structure contains three degrees of freedom: its out-of-plane displacement $w$ and the surface slopes $w,_x$ and $w,_y$. The nodes are uniformly distributed on the square spacecraft and triangular elements between the nodes are defined. The finite element mesh is shown in figure 5.4. The 4 corner forces from the thrusters are also shown. The nodal degrees of freedom are defined in figure 5.5.

Shape functions are defined over each element in order to reconstruct the out-of-plane coordinate of a point on the structure. Different functions are used to reconstruct

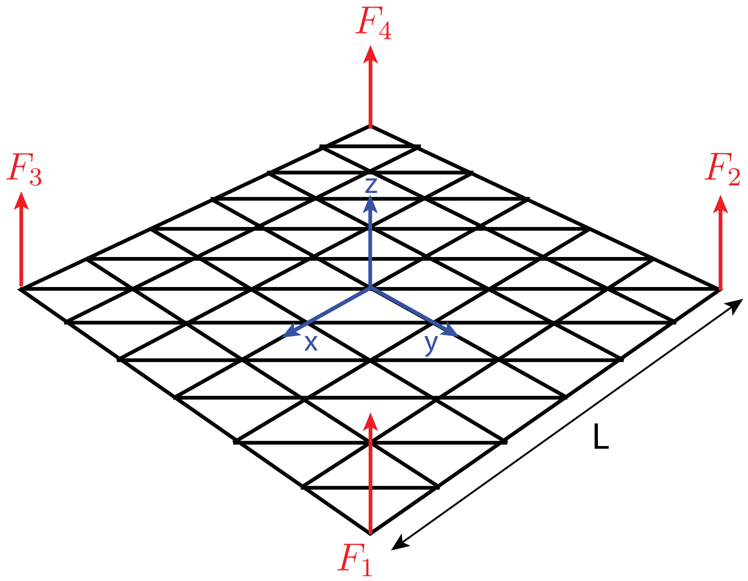


Figure 5.4: Definition of the element layout on the plate-like spacecraft as well as the 4 corner forces.

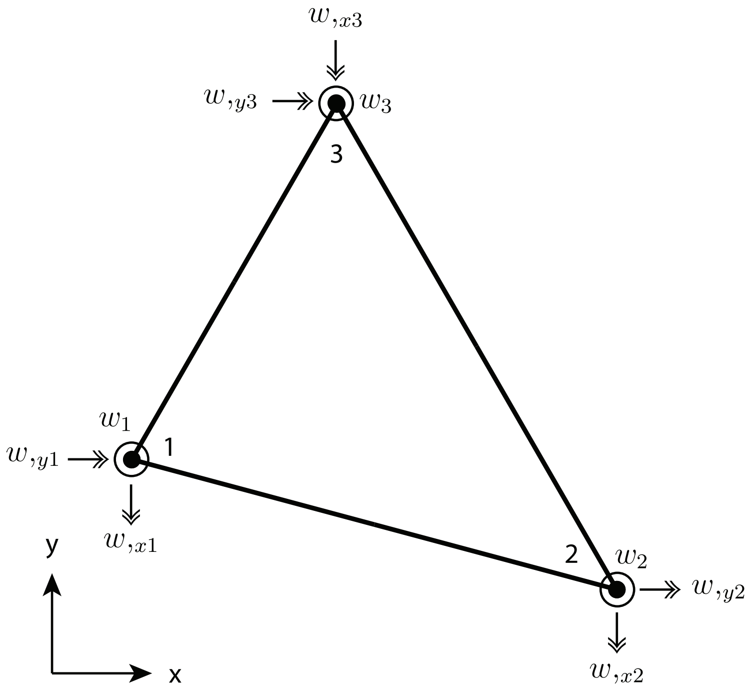


Figure 5.5: Definition of the triangular element and its degrees of freedom.

the mass and stiffness matrix. A simple 3rd order polynomial is used to build the mass matrix (equation 5.13). The out-of-plane displacement within an element is written as:

$$w = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 + a_6 x^3 + a_7 x^2 y + a_8 xy^2 + a_9 y^3 \quad (5.26)$$

where the coefficients $a_k$ are computed from the nodal degrees of freedom. Because only 9 d.o.f. are available and 10 coefficients need to be computed, we impose the following relation to enforce continuity of the normal slopes across elements:

$$a_7 + a_8 = 2(a_6 + a_9) \quad (5.27)$$

As mentioned in [21], this formulation can be singular for some element geometries. The simple element layout shown in figure 5.4 results in a well-posed formulation and is used to compute the mass matrix in equation 5.21. It is also used to compute the vector of external forces in equation 5.23.

Discrete Kirchhoff Triangular Elements (DKT) are used to compute the stiffness matrix. DKT is considered one of the best three-node plate elements [6, 21]. The complete formulation can be found in [21]. After assembling the different element contributions, the classical stiffness matrix, equation 5.15, fully defines the full model matrix equation 5.22.

**Reduced Order Model**

The equation of motion (equation 5.24) while linear can contain a large number of degrees of freedom: 3 × (Number of nodes) + 1. As shown in section 5.4, the matrices used in the Kalman filter are even larger (over double the number of d.o.f.) which leads to slow computations. This can be problematic especially if such an algorithm is to be programmed on the on-board computer of a spacecraft.

The finite element model aims to capture the vibrations of the spacecraft. In the current form, it contains high order vibration modes which in most cases are not necessary. Indeed, these high frequency modes are usually not excited, especially in the case of slow maneuvers.

A Craig and Bampton model reduction is performed on equation 5.24. This methodology replaces the classical element-based model with a super-element. The degrees of freedom of the super-element are a subset of d.o.f. from the original formulation and a set of modal amplitudes corresponding to the eigenmodes of the structure the user wishes to retain [32, 68].

Let $\mathbf{q}_B$ be the retained degrees of freedom and $\mathbf{q}_I$ the other (also called internal) degrees of freedom. The equation of motion can be re-written by separating the contribution of each set:

$$\begin{bmatrix} \mathbf{M}_{BB} & \mathbf{M}_{BI} \\ \mathbf{M}_{IB} & \mathbf{M}_{II} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_B \\ \ddot{\mathbf{q}}_I \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{BB} & \mathbf{K}_{BI} \\ \mathbf{K}_{IB} & \mathbf{K}_{II} \end{bmatrix} \begin{bmatrix} \mathbf{q}_B \\ \mathbf{q}_I \end{bmatrix} = \begin{bmatrix} \mathbf{F}_B \\ \mathbf{0} \end{bmatrix} \tag{5.28}$$

where $\mathbf{F}_B$ is the vector of possible applied forces or reaction forces. In the presented formulation, $\mathbf{q}_B$ contains the out-of-plane displacement of the corner nodes where the thruster forces are applied, and the out-of-plane displacement and the slope around $y$ of the central node that are set to zero to eliminate redundant rigid-body motions as explained in the previous sub-section. All other d.o.f. are contained in $\mathbf{q}_I$.

The vector of internal d.o.f. $\mathbf{q}_I$ can be written as a superposition of two solutions: the static response from the retained d.o.f. $\mathbf{q}_B$ and the dynamic response calculated by fixing $\mathbf{q}_B$:

$$\mathbf{q}_I = \mathbf{q}_I^S + \mathbf{q}_I^D \tag{5.29}$$

where $\mathbf{q}_I^S$ is the static response:

$$\mathbf{q}_I^S = -\mathbf{K}_{II}^{-1}\mathbf{K}_{IB}\mathbf{q}_B \tag{5.30}$$

and $\mathbf{q}_I^D$ is the dynamic response calculated by fixing $\mathbf{q}_B$ to $\mathbf{0}$. It solves the eigenvalue problem:

$$\mathbf{K}_{II}\mathbf{q}_I^D = \omega^2\mathbf{M}_{II}\mathbf{q}_I^D \tag{5.31}$$

The solutions of equation 5.31 can be set into a matrix $\mathbf{\Phi}_I$. Each solution is scaled such that:

$$\mathbf{\Phi}_I^T\mathbf{K}_{II}\mathbf{\Phi}_I = \begin{bmatrix} \omega_1^2 & & 0 \\ & \ddots & \\ 0 & & \omega_{nI}^2 \end{bmatrix} = \mathbf{\Omega}_I \tag{5.32}$$

$$\mathbf{\Phi}_I^T\mathbf{M}_{II}\mathbf{\Phi}_I = \mathbf{I} \tag{5.33}$$

The dynamic contribution of $\mathbf{q}_I$ can be written as:

$$\mathbf{q}_I^D = \mathbf{\Phi}_I\mathbf{a}_I \tag{5.34}$$

with $\mathbf{a}_I$ the vector of amplitude of each eigenmodes. Instead of adding all possible frequency contributions, only a subset of eigenmodes is chosen:

$$\mathbf{\Phi}_m = [\mathbf{\Phi}_I^{(1)} \quad \dots \quad \mathbf{\Phi}_I^{(m)}] \tag{5.35}$$

$$\mathbf{a}_m = [a_I^{(1)} \quad \dots \quad a_I^{(m)}]^T \tag{5.36}$$

The initial equations of motion can be then re-written in terms of the retained degrees of freedom and the chosen vibration mode amplitudes:

$$\begin{bmatrix} \overline{\mathbf{M}}_{BB} & \mathbf{M}_{Fm} \\ \mathbf{M}_{mF} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_B \\ \ddot{\mathbf{a}}_m \end{bmatrix} + \begin{bmatrix} \overline{\mathbf{K}}_{BB} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Omega}_m \end{bmatrix} \begin{bmatrix} \mathbf{q}_B \\ \mathbf{a}_m \end{bmatrix} = \begin{bmatrix} \mathbf{F}_B \\ \mathbf{0} \end{bmatrix} \tag{5.37}$$

with,

$$\overline{\mathbf{M}}_{BB} = \mathbf{M}_{BB} - \mathbf{M}_{BI}\mathbf{K}_{II}^{-1}\mathbf{K}_{IB} - \mathbf{K}_{BI}\mathbf{K}_{II}^{-1}\mathbf{M}_{IB} + \mathbf{K}_{BI}\mathbf{K}_{II}^{-1}\mathbf{M}_{II}\mathbf{K}_{II}^{-1}\mathbf{K}_{IB} \tag{5.38}$$

$$\mathbf{M}_{mF} = \mathbf{M}_{Fm}^{T} = \mathbf{\Phi}_m^{T}\left(\mathbf{M}_{IB} - \mathbf{M}_{II}\mathbf{K}_{II}^{-1}\mathbf{K}_{IB}\right) \tag{5.39}$$

$$\overline{\mathbf{K}}_{BB} = \mathbf{K}_{BB} - \mathbf{K}_{BI}\mathbf{K}_{II}^{-1}\mathbf{K}_{IB} \tag{5.40}$$

The initial vector of degrees of freedom can be reconstructed as follows:

$$\begin{bmatrix} \mathbf{q}_B \\ \mathbf{q}_I \end{bmatrix} = \mathbf{L} \begin{bmatrix} \mathbf{q}_B \\ \mathbf{a}_m \end{bmatrix} \tag{5.41}$$

where

$$\mathbf{L} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{K}_{II}^{-1}\mathbf{K}_{IB} & \mathbf{\Phi}_m \end{bmatrix} \tag{5.42}$$

## 5.4 Kalman Filter Formulation

A Kalman filter [14] is a recursive filter that uses a dynamic model to predict the behavior of the state variables, in combination with measurements of these variables to correct the prediction. This filter is optimal in the sense that it minimizes the RMS error of the solution considering the noise of the sensors and errors of the dynamic model.

We present a brief summary of the equations used in the filter and define the main matrices. Details can be found in [14].

The first step in the development of a Kalman filter is a recursive dynamic model that captures the actual shape of the structure in the following form:

$$\mathbf{x}_{k+1} = \mathbf{A}_d\mathbf{x}_k + \mathbf{w}_k \tag{5.43}$$

where $\mathbf{x}_k$ is the state vector at time $t_k$, $\mathbf{A}_d$ is the state transition matrix, and $\mathbf{w}_k$ is the process noise which characterizes the errors of the model. This process noise is assumed to be zero-mean, gaussian with a covariance matrix $\mathbf{Q}_d$.

In the Kalman filter, the predicted state $\hat{\mathbf{x}}$ propagates as the mean of the dynamic model:

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}_d\hat{\mathbf{x}}_k \tag{5.44}$$

The covariance of the state $\mathbf{P}_k$ can also be estimated and propagated:

$$\mathbf{P}_{k+1} = \mathbf{A}_d \mathbf{P}_k \mathbf{A}_d^T + \mathbf{Q}_d \tag{5.45}$$

The measurements at time $t_k$ can be written as a linear combination of the entries of the state vector:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{5.46}$$

where $\mathbf{H}$ is the measurement matrix and $\mathbf{v}_k$ is the noise of the measurement, assumed to be zero-mean, gaussian with covariance $\mathbf{R}$.

After each measurement, the predicted state $\hat{\mathbf{x}}_k$ obtained from the propagation equation 5.44 can be corrected using the formula:

$$\hat{\mathbf{x}}_k \Leftarrow \hat{\mathbf{x}}_k + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k\right) \tag{5.47}$$

where $\mathbf{K}_k$ is the Kalman gain, an optimal gain calculated from the covariance of the state and measurement:

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}^T \left(\mathbf{H}\mathbf{P}_k \mathbf{H}^T + \mathbf{R}\right)^{-1} \tag{5.48}$$

The covariance of the state $\mathbf{P}_k$ also needs to be updated after a measurement using the equation:

$$\mathbf{P}_k \Leftarrow \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}\right) \mathbf{P}_k \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}\right)^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \tag{5.49}$$

The updated state $\hat{\mathbf{x}}_k$ and covariance $\mathbf{P}_k$ are then used as inputs of the propagation equations 5.44 and 5.45. This creates a loop whose input is the measurement $\mathbf{z}_k$ and output is the predicted state $\hat{\mathbf{x}}_k$. Figure 5.6 summarizes the Kalman filter loop.

### Discrete Time State Model

The discrete time state model is generated from the finite element model. The detailed derivation of the equations can be found in [14, 20]. [20] specifically details the formulation of a Kalman filter from a finite element model. We recall the dynamic equation of the finite element model (equation 5.24):

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F}_a \tag{5.50}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{q}$ is the vector of degrees of freedom, $\mathbf{K}$ is the stiffness matrix, and $\mathbf{F}_a$ is the vector of nodal forces. These matrices and vectors are generated from the floating frame of reference formulation 5.3 and further reduced to produce a reduced order model.

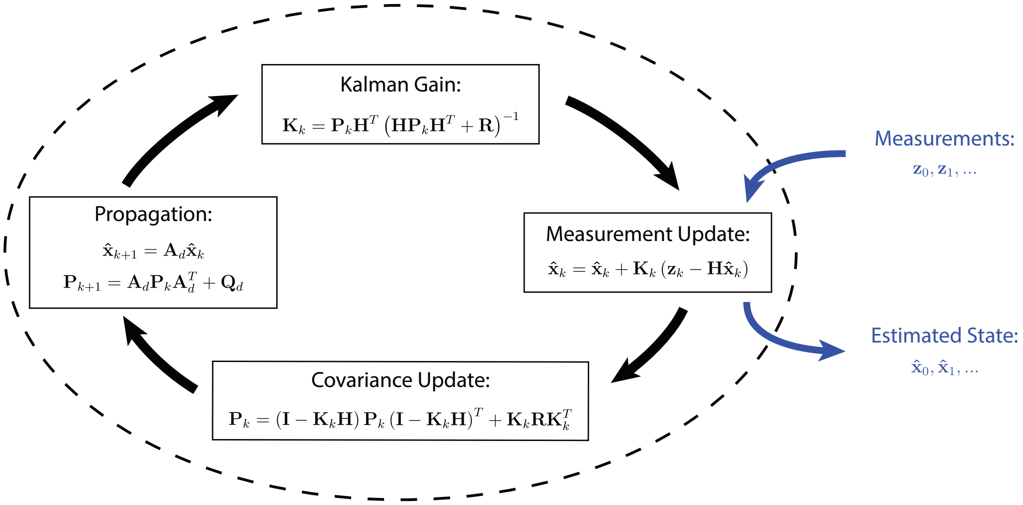

Kalman Gain:
$\mathbf{K}_k = \mathbf{P}_k\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_k\mathbf{H}^T + \mathbf{R}\right)^{-1}$
Measurements:
$\mathbf{z}_0, \mathbf{z}_1, \ldots$
Propagation:
$\hat{\mathbf{x}}_{k+1} = \mathbf{A}_d\hat{\mathbf{x}}_k$
$\mathbf{P}_{k+1} = \mathbf{A}_d\mathbf{P}_k\mathbf{A}_d^T + \mathbf{Q}_d$
Measurement Update:
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k\right)$
Estimated State:
$\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \ldots$
Covariance Update:
$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}\right)\mathbf{P}_k\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}\right)^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T$

Figure 5.6: Kalman filter loop.

The force vector mostly contains zeros. We can rewrite it as $\mathbf{F}_a = \mathbf{F}_d\tilde{\mathbf{F}}$ where $\mathbf{F}_d$ is commonly called the control distribution matrix and $\tilde{\mathbf{F}} = [F_1 \quad F_2 \quad F_3 \quad F_4]^T$ is the vector of force amplitudes. The forces are defined in figure 5.4.

We first modify equation 5.50 to create the first order ODE:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \tilde{\mathbf{w}} \tag{5.51}$$

where $\tilde{\mathbf{w}}$ is the noise or uncertainty of the model and

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \\ \tilde{\mathbf{F}} \end{bmatrix} \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\mathbf{M}^{-1}\mathbf{K} & \mathbf{0} & \mathbf{M}^{-1}\mathbf{F}_d \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{5.52}$$

Note that the forces are included in the state vector $\mathbf{x}$ (as opposed to being included as a control term as in [20]). A simple zeroth order integrator characterizes their behavior (last row of $\mathbf{A}$) and their time derivative $\dot{\tilde{\mathbf{F}}}$ is assumed to be pure zero-mean gaussian noise (contained in the last rows of $\tilde{\mathbf{w}}$).

Equation 5.51 can be integrated to create the propagation equation 5.44 with:

$$\mathbf{A}_d = e^{\mathbf{A}dt} \tag{5.53}$$

where $dt = t_{k+1} - t_k$.

The covariance of the process noise $\mathbf{Q}_d$ can also be calculated from the covariance of $\tilde{\mathbf{w}}$. We assume that the noise of the model comes from the applied forces only.

The covariance of the process noise $\tilde{\mathbf{w}}$ can be written as:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{F}_d \\ \mathbf{0} \end{bmatrix} \mathbf{Q}_F \begin{bmatrix} \mathbf{0} & \mathbf{M}^{-1}\mathbf{F}_d & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I}_{4\times4} \end{bmatrix} \mathbf{Q}_{\dot{F}} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I}_{4\times4} \end{bmatrix} \tag{5.54}$$

where $\mathbf{Q}_F$ is the covariance matrix of the forces and $\mathbf{Q}_{\dot{F}}$ is the covariance matrix of the forces time derivatives.

$\mathbf{Q}_F$ is calculated from the maximum amplitude of the applied forces by the thrusters. The amplitude of $\mathbf{Q}_{\dot{F}}$ is a tuning parameter that characterizes how much the estimated forces from the state vector can be modified by the measurement. A small value restrains the forces to remain fairly constant in time posing the risk of missing the different actuations. A high value lets the noisy measurement dictate the force amplitude which could lead to a noisy estimation of the state vector. This effect will be studied in the next section.

The covariance of the process noise $\mathbf{Q}_d$ is calculated from the integration of equation 5.51 from $t_k$ to $t_{k+1}$. A first order approximation will be used such that:

$$\mathbf{Q}_d = dt\mathbf{Q} \tag{5.55}$$

### Measurement Equations

The sun sensors measure the local angles of the structure from the sun axis. We assume that the Sun is a point source at infinity along the $Z$-axis (inertial reference frame). The sensors are aligned on the structural coordinate system, as shown in figure 5.7. The angles measured by sensor $S$ are derived in chapter 2, equations 2.24 and 2.25:

$$\tan\alpha_S = \frac{\mathbf{Z}\cdot\left.\frac{\partial\mathbf{r}}{\partial y}\right|_S}{\mathbf{Z}\cdot\mathbf{n}_S}\frac{\|\mathbf{n}_S\|}{\left\|\left.\frac{\partial\mathbf{r}}{\partial y}\right|_S\right\|} \tag{5.56}$$

$$\tan\beta_S = -\frac{\mathbf{Z}\cdot\left.\frac{\partial\mathbf{r}}{\partial x}\right|_S}{\mathbf{Z}\cdot\mathbf{n}_S}\frac{\|\mathbf{n}_S\|}{\left\|\left.\frac{\partial\mathbf{r}}{\partial x}\right|_S\right\|} \tag{5.57}$$

where $\mathbf{n} = \frac{\partial\mathbf{r}}{\partial x}\times\frac{\partial\mathbf{r}}{\partial y}$ is the normal to the structure.

The position of the sensor can be written using the model formulation detailed in 5.3:

$$\mathbf{r}_S = \mathbf{A}_x\left(\begin{bmatrix} x_S \\ y_S \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \tilde{\mathbf{N}}^S \end{bmatrix}\mathbf{q}_f\right) \tag{5.58}$$

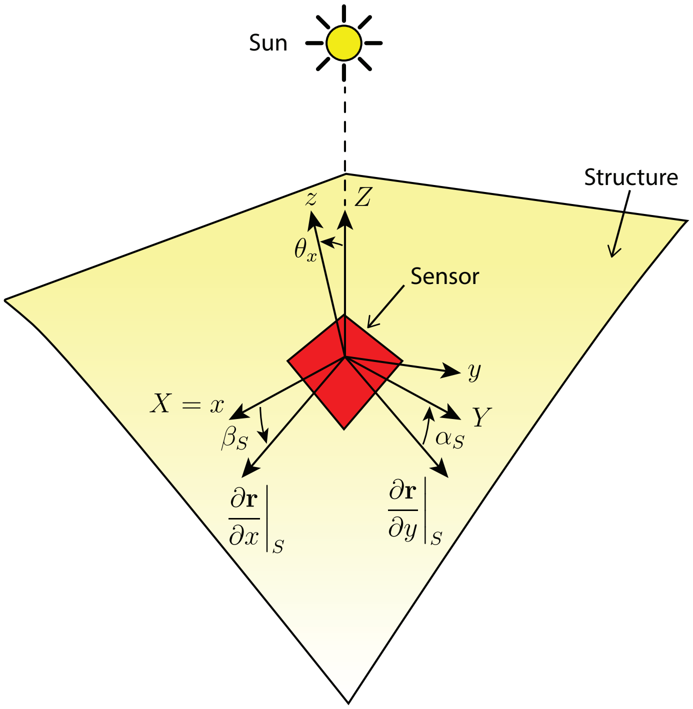

Sun
Structure
$z$
$Z$
$\theta_x$
Sensor
$y$
$X = x$
$Y$
$\beta_S$
$\alpha_S$
$\left.\frac{\partial \mathbf{r}}{\partial x}\right|_S$
$\left.\frac{\partial \mathbf{r}}{\partial y}\right|_S$

Figure 5.7: Definition of the angles $\alpha$ and $\beta$ measured by a sensor. The sunlight comes from the $Z$-direction (intertial reference frame) and the sensor is aligned towards the x and y axes of the structure and follows its deformation.

where $\mathbf{A}_x$ is the rotation matrix around the $x$-axis:

$$\mathbf{A}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \tag{5.59}$$

Let us assume that there is a node at the location of the sensor. The derivatives of the position of the sensor are:

$$\left.\frac{\partial \mathbf{r}}{\partial x}\right|_S = \mathbf{A}_x \begin{bmatrix} 1 \\ 0 \\ w,_{xS} \end{bmatrix} \tag{5.60}$$

$$\left.\frac{\partial \mathbf{r}}{\partial y}\right|_S = \mathbf{A}_x \begin{bmatrix} 0 \\ 1 \\ w,_{yS} \end{bmatrix} \tag{5.61}$$

These can be plugged back into the measurement equations 5.56 and 5.57. Assuming small deformations, the measurement equations can be written as functions of $\alpha_S$

and $\beta_S$ directly:

$$\alpha_S = \theta_x + w,_{yS} \tag{5.62}$$

$$\beta_S = -w,_{xS} \tag{5.63}$$

The two measurements for all the sensors can be concatenated to create a linear measurement system:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{5.64}$$

where $\mathbf{z}_k$ is the measurement vector, concatenation of all measured angles $\alpha_S$ and $\beta_S$, $\mathbf{H}$ is the measurement matrix, $\mathbf{x}_k$ is the state vector of the Kalman filter defined in equation 5.52, and $\mathbf{v}_k$ is the zero-mean gaussian noise of the sensors.

### Observability of the State

To ensure that the Kalman filter is well behaved, it is necessary to make sure that the system is observable. This means that all entries of the state vector can be inferred from a sequence of measurements. Different methods exist to measure the observability of the system such as checking the singular values of the observability matrix or the observability grammian [97]. Generally, the Kalman filter system is observable if the observability matrix has full rank [79]. Let $n$ be the size of the state vector. Then the observability matrix is defined by:

$$\mathbf{O}_n = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}\mathbf{A}_d \\ \mathbf{H}\mathbf{A}_d^2 \\ \vdots \\ \mathbf{H}\mathbf{A}_d^{n-1} \end{bmatrix} \tag{5.65}$$

Note that the number of columns of the matrix is $n$, hence the observability matrix $\mathbf{O}_n$ needs to have a rank equal to $n$.

It is sufficient that $\text{rank}(\mathbf{O}_\nu) = n$ with $\nu \leq n$. The minimum $\nu$ is called the observability index and corresponds to the minimum number of independent measurements required to infer the whole state vector. This allows for the observability criterion to be calculated iteratively which is helpful when the dimension of the state vector $n$ is large and numerical errors can quickly interfere with the rank calculation.

For the structure presented in the section 5.5, a $25 \times 25$ m structure with a $25 \times 25$ grid of nodes is used to generate the finite element model. The model is reduced to only keep the first 30 vibration modes, the rigid-body angle around the $x$-axis, and

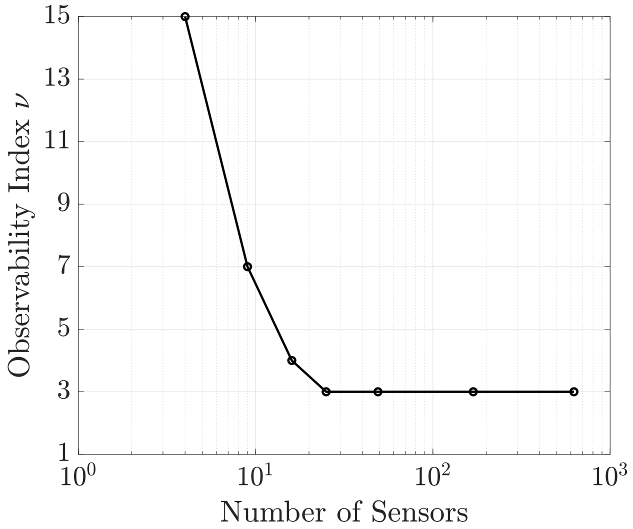


Figure 5.8: Observability index in function of the number of sensors. The reduced order model retained 30 modes of vibration. Sensors are distributed linearly on the structure in both directions. The first point corresponds to a 2 × 2 array of sensors; the last point to a 25 × 25 one.

the necessary degrees of freedom associated with external forces. The observability index was calculated for 8 different sensor configurations: $1 \times 1$, $2 \times 2$, $3 \times 3$, $4 \times 4$, $5 \times 5$, $7 \times 7$, $13 \times 13$, and $25 \times 25$.

The results of the study are shown in figure 5.8. The system is not observable with only 1 sensor placed on the structure as the system cannot differentiate between the 4 modes of actuation created by the 4 thrusters. The system is observable for all other configurations, and the larger the number of sensors, the faster the system is at observing the state vector. The observability index eventually converges to 3. This corresponds to the number of iteration needed to propagate the force amplitudes (last entries of the state vector) to the measured angles.

It is important to note that some assumptions made in the finite element model (section 5.3) are actually crucial when it comes to observability. If translation modes were to be included, the system set up for the Kalman filter becomes ill-posed as any translation will be unobservable. It is easy to show that the first columns of $\mathbf{O}_\nu$, associated with the translation of the structure, are zero, hence reducing the order of the matrix. This is due to the fact that the measurements are invariant by translation of the structure. For the same reason, any rotation around the Sun axis ($Z$-axis as defined in the previous sub-section) is also not observable.

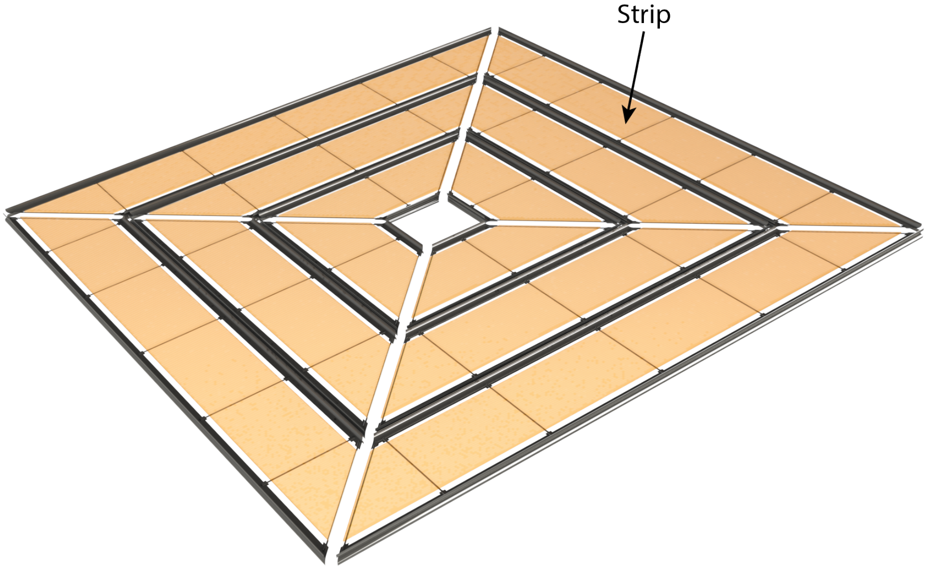


Figure 5.9: Architecture of a 1.7 m concept of the Space Solar Power Spacecraft structure [30]

## 5.5 Application to a Proposed Space Mission: the Caltech Space Solar Power Project (SSPP)

The Kalman filter detailed in section 5.4 was applied to a model of the Space Solar Power Satellite under development at Caltech. More details about this spacecraft can be found in [3, 30, 31, 35, 45, 53]. The aim of this satellite is to collect energy from sunlight in space, convert it to radio frequency and beam it back to Earth, eventually creating a solar power station.

Figure 5.9 shows the architecture of the spacecraft. It is composed of 4 quadrants made of multiple strips. The yellow area is filled with solar panels on one side and antennas on both sides while the black elements are carbon fiber longerons which provide bending stiffness to the structure in order to maintain its planarity. Different sizes of the spacecraft have been considered ranging from a few meters up to 60 m side length.

While the structure has some bending stiffness and is very lightweight, large concepts can still deform to tens of centimeters out-of-plane, due to the thruster forces required to maneuver the satellite [53]. While small compared to the size of the satellite, these deformations are large enough to interfere with the antenna array placed on the structure. Errors in the position of the antennas need to remain below 1/10th of the operating wavelength. If larger, they can however be accounted for by locality adding a phase shift to each antenna. The shape of the spacecraft needs to be measured to less than 1/10th of the antenna array wavelength (3 cm for a 10 GHz array) for this system to work.

A simple model of this spacecraft under the loads required to maintain its attitude is studied. Thrusters at each corner of the structure can fire up or down along the $z$-axis to generate the moments required for the maneuvers. The main axis of rotation is the $x$-axis as shown in figure 5.4 which is aligned with the side of the spacecraft.

We study a $25 \times 25$ m structure using the model in section 5.3. A finite element model is generated using a grid of $25 \times 25$ nodes. Its bending stiffness is derived from a detailed model of the SSPP structure, which has a bending stiffness of approximately 14 Nm$^2$ [69]. The mass density, extrapolated from small scale models of the structure, is equal to 420 g/m$^2$.

An array of sensors is uniformly distributed on the structure (it is not optimized to fit on the a structure similar to figure 5.9). The noise properties of the sensors, necessary for the Kalman filter, are derived from an actual sensor manufactured using a quad-photodiode (see chapter 3). The $3\sigma$ noise of the sensors is assumed equal to 0.5° for both measured angles.

### Actual Deformation of the Structure

In order to implement the Kalman filter, the input measurement of the sensors need to be simulated. The complete shape of the spacecraft under the thruster loads required for maneuvering is computed. The measurement is calculated by extracting the angles of the structure at the sensor locations and by adding gaussian noise.

An approach similar to the discrete time state model used for the Kalman filter is used to simulate the actual shape of the spacecraft (see section 5.4), and a finite element model using the Floating Frame of Reference Formulation and the local small deformation plate model is created. This model is not reduced to capture a more complete behavior of the vibrating spacecraft.

A first order ODE system is then developed to solve the equation of motion of the model (equation 5.24):

$$\begin{bmatrix} \dot{\mathbf{q}}(t) \\ \ddot{\mathbf{q}}(t) \end{bmatrix} = \mathbf{A}_t \begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix} + \mathbf{B}_t \tilde{\mathbf{F}}_{\text{actual}}(t) \tag{5.66}$$

with,

$$\mathbf{A}_t = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \quad \text{and} \quad \mathbf{B}_t = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{F}_d \end{bmatrix} \tag{5.67}$$

where $\mathbf{M}$ and $\mathbf{K}$ are the non-reduced mass and stiffness matrices, $\mathbf{q}$ is the vector of degrees of freedom of the finite element model, $\mathbf{F}_d$ is the control distribution matrix introduced in 5.4, and $\tilde{\mathbf{F}}_{\text{actual}}$ are the actual forces imposed in the structure. $\mathbf{C} = 10^{-4}\mathbf{K}$ is a Rayleigh stiffness damping matrix (artificial damping) that prevents the contribution of high order vibration modes which are induced by sudden changes in external loads which does not happen in reality but happen in discrete time simulations.

Equation 5.66 can be integrated in time to generate a discrete time state model. More details of the derivation can be found in [20].

$\tilde{\mathbf{F}}_{\text{actual}}$ is a column vector of four entries corresponding to the 4 thruster forces applied at the corner of the structure. They are calculated from the optimal maneuvers of the spacecraft [53]. They consist of 30 s impulses of 3.3 mN (millinewtons) amplitude that eventually rotate the spacecraft around the $x$-axis. A first impulse starts the maneuver while a second, about 1 hour and 10 mins later terminates it. Figure 5.10 shows the profile of $F_1$ defined in figure 5.4. Note that $F_1 = F_2 = -F_3 = -F_4$ to generate a large rotation around the $x$-axis only. The rigid-body angle $\theta_x$ is plotted in figure 5.11. An initial angle and velocity are used to match the optimal attitude of the spacecraft before the maneuver [53].

Figure 5.12 shows the $Z$-coordinate (inertial reference frame) of the corner node where $F_1$ is applied. We see that its behavior is mostly dictated by the rigid rotation of the structure. Figure 5.13 shows the $z$-coordinate (local reference frame) of the same corner node. We can better appreciate the amplitude of the vibrations which are in the order of a few centimeters. Note that an initial vibration of the structure was set up by applying 2 opposite, 5 sec impulses of 6.6 mN at the start of the simulation (see figure 5.10).

Finally, we can notice the coupling of the dynamics of the rigid-body angle $\theta_x$ and the vibration of the structure by zooming into figure 5.11. Figure 5.14 shows the behavior of this angle at the end of the maneuver. We clearly see oscillations in $\theta_x$. They are relatively small due to the large inertia of the structure.

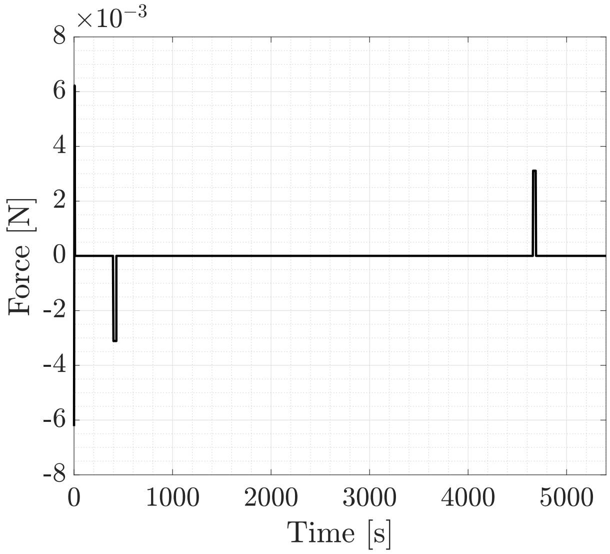


Figure 5.10: External force applied at one corner of the structure. Two short impulses at $t = 0$ have the sole purpose of introducing vibrations in the structure. The longer, 30s impulses force the satellite to rotate around the $x$-axis.

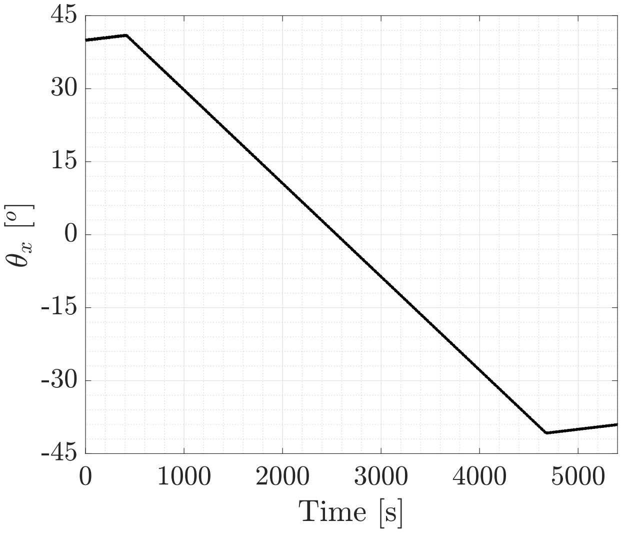


Figure 5.11: Rigid body rotation of the spacecraft around the $x$-axis.

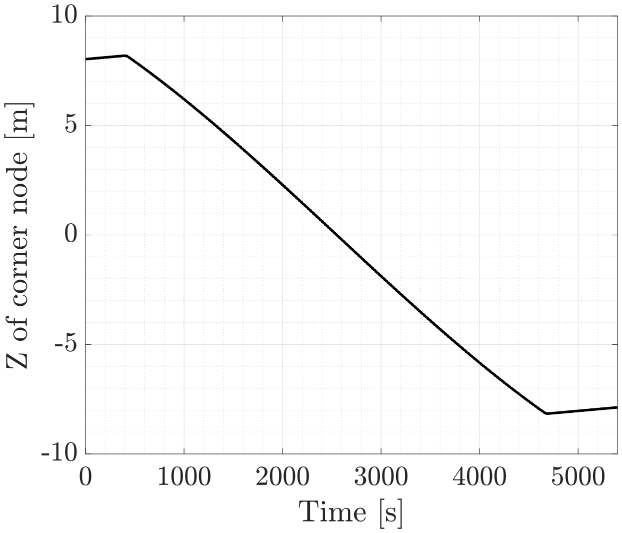


Figure 5.12: $Z$-coordinate of the corner node of the structure (inertial reference frame).

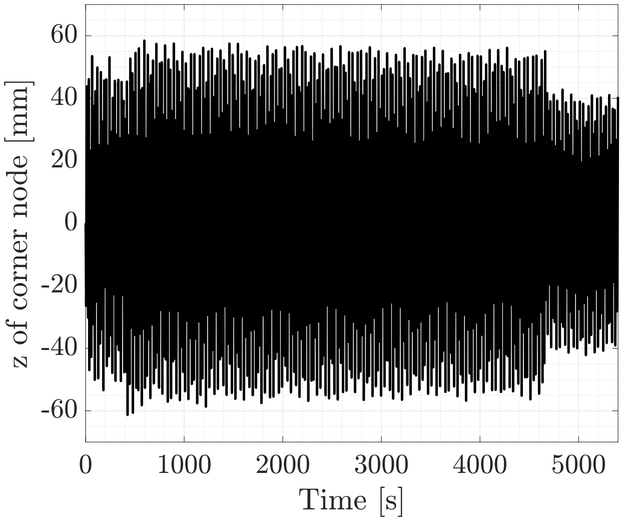


Figure 5.13: $z$-coordinate of the corner node of the structure (local reference frame).

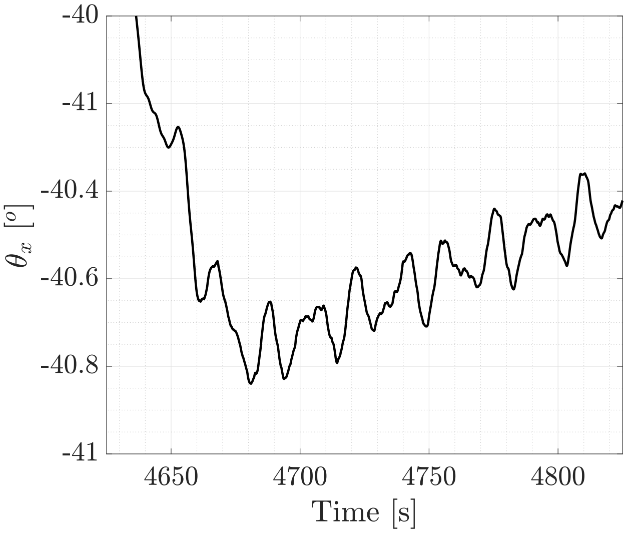


Figure 5.14: Rigid body angle of the spacecraft around the $x$-axis. Zoom of figure 5.11 around the second actuation.

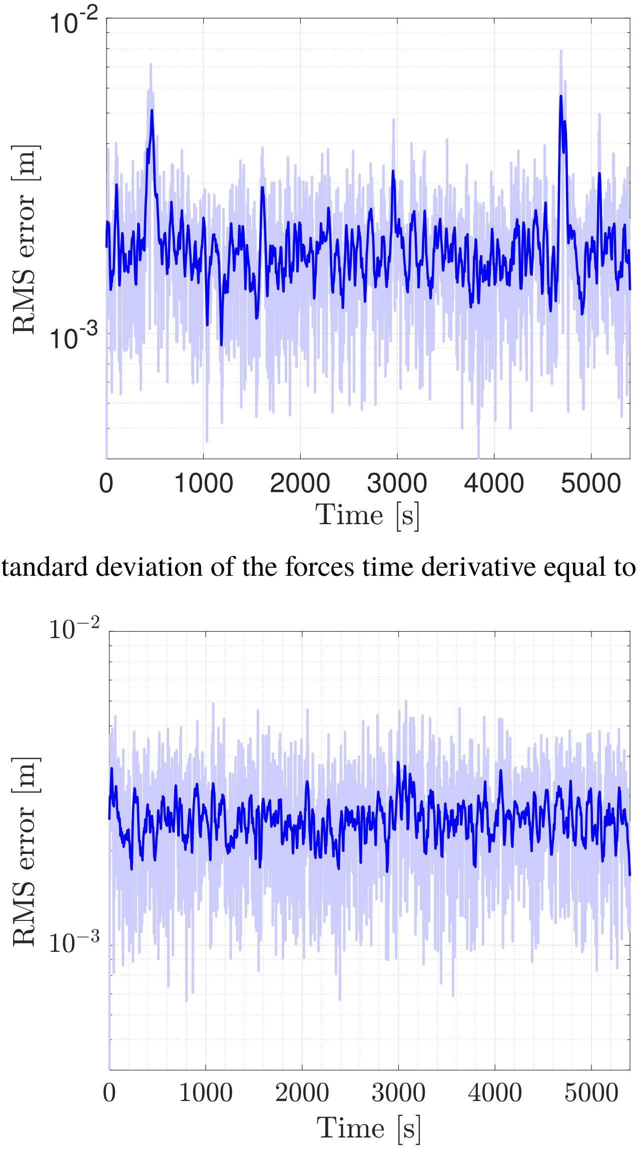
### Tuning the Process Noise of the Kalman Filter

As shown in section 5.4, the dynamic model of the spacecraft is associated with some uncertainty. This is captured by the process noise which, as detailed in equation 5.54 is the sum of two contributions: the uncertainty in the force amplitude $\mathbf{Q}_F$ and the uncertainty in their time derivative $\mathbf{Q}_{\dot{F}}$.

As the forces can vary by up to about 3 mN, a gaussian noise of 1 mN standard devation is chosen such that $\mathbf{Q}_F = 0.001^2 \times \mathbf{I}_{4\times4}$ [N$^2$].

The second contribution $\mathbf{Q}_{\dot{F}}$ is left as a tuning parameter. To show the effect of high and low values, we conduct two simulations of the Kalman filter. A grid of $7 \times 7$ sensors is used and we plot the RMS error between the estimated shape (output of the filter) and the actual shape of the structure presented in the previous sub-section. The results are shown in figure 5.15. For small values of $\mathbf{Q}_{\dot{F}}$ (figure 5.15a) the error suddenly increases when force impulses are applied. This is to be expected as a small noise indicates to the filter that the forces should vary slowly effectively increasing the response time of the system. When no force is applied, however, this constraint reduces the impact of the sensor noise onto the results, hence the RMS drops to relatively low values. When $\mathbf{Q}_{\dot{F}}$ is larger (figure 5.15b), the filter is faster to respond such that there is no difference whether a force is applied or not. The RMS is however higher on average.

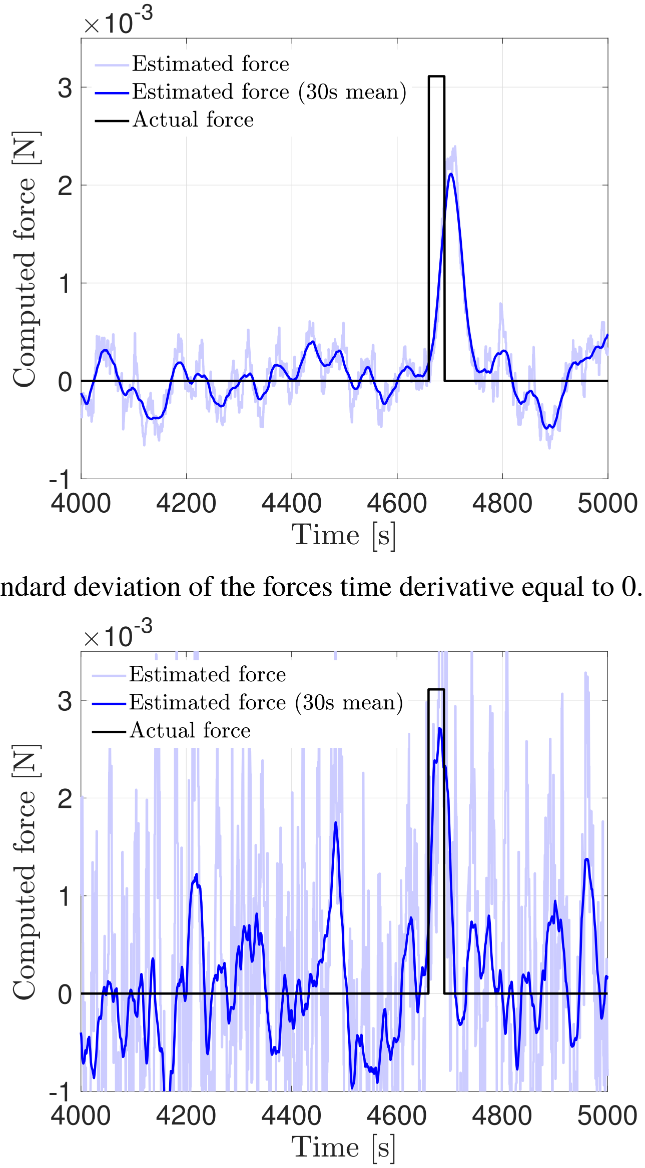
These results can also be inferred by looking at the amplitude of the estimated forces



(a) Standard deviation of the forces time derivative equal to 0.1 mN/s.

(b) Standard deviation of the forces time derivative equal to 1 mN/s.

Figure 5.15: RMS error of the estimated shape for different standard deviations of the forces time derivative used in the process noise covariance matrix.



(a) Standard deviation of the forces time derivative equal to 0.1 mN/s.

(b) Standard deviation of the forces time derivative equal to 1 mN/s.

Figure 5.16: Estimated corner force for different standard deviations of the forces time derivative used in the process noise covariance matrix.

in the state vector. Figure 5.16 shows the estimation of the corner force $F_1$ for the two different values of $\mathbf{Q}_{\dot{F}}$. When the process noise is small, the filter is slow to converge, but stays close to its converged estimation. A large noise leads to the opposite conclusion.

Unless specified, a standard deviation of 1 mN/s ($\mathbf{Q}_{\dot{F}} = 0.001^2 \times \mathbf{I}_{4\times4}$ [N$^2$/s$^2$]) is chosen. Depending on the operation of the satellite, an optimized value can be calculated. The process noise could also be actively controlled and set to higher values before firing the thrusters.

### Response of the Kalman Filter Using a $7 \times 7$ Array of Sensors

We study the accuracy of the Kalman filter introduced in section 5.4. The filter tries to reconstruct the simulated shape of the spacecraft presented in a previous sub-section. The simulated angle measurements are inputted into the Kalman filter which outputs the estimated shape of the spacecraft.

The the state vector $\mathbf{x}$ (see equation 5.52) contains the degrees of freedom of the reduced finite element model plus the 4 external forces. The first 30 vibration modes of the structure are retained in the reduced model. The number of d.o.f. of the FE model is equal to 35 which corresponds to:

1. the 30 vibration mode amplitudes
2. the displacement of the 4 corner nodes where the forces are applied (this is necessary to apply the boundary conditions)
3. the large rigid-body angle $\theta_x$.

The size of the state vector is equal to 74 (double the number of d.o.f. plus the 4 corner forces).

A $7 \times 7$ array of sun sensors is placed on the structure. They are equally spaced from each other to cover the whole spacecraft.

Figure 5.17 shows the $Z$-coordinate (inertial reference frame) of the corner node where $F_1$ is applied. The overall trend of the curve is similar to figure 5.12 and is dominated by the rigid angular motion. Figure 5.18 shows the error between the estimated and true $Z$-coordinate of that point. The noise of the estimated shape of the structure can clearly be seen which is the result of the noisy measurements. It is stable in time with an amplitude of about a centimeter and is not affected by the force impulses.

The error in the $z$-coordinate (local reference frame) can also be computed and is plotted in figure 5.19. We see that the error is higher than in figure 5.18. This has to do with the fact that the rigid-body angle is not perfectly computed either (see figure 5.20) and eventually compensates some of this error.

In order to better study the overall accuracy of the estimated shape, the RMS error across the surface is computed. The error corresponds to the distance between each

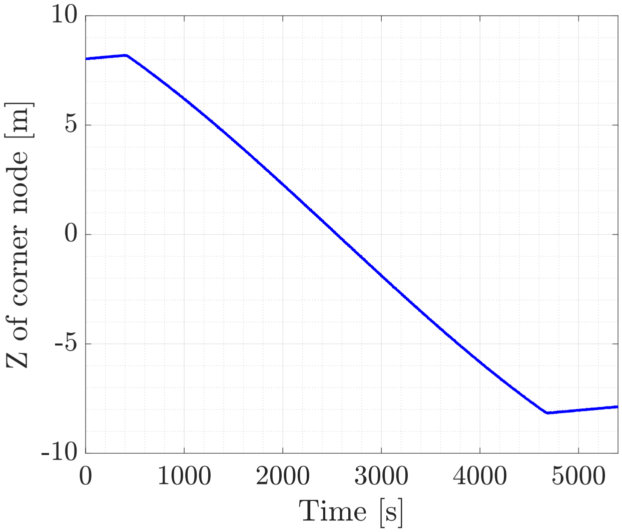


Figure 5.17: Evolution of the $Z$-coordinate (inertial reference frame) of the estimated shape of the spacecraft calculated by the Kalman Filter using a $7 \times 7$ sensor array.

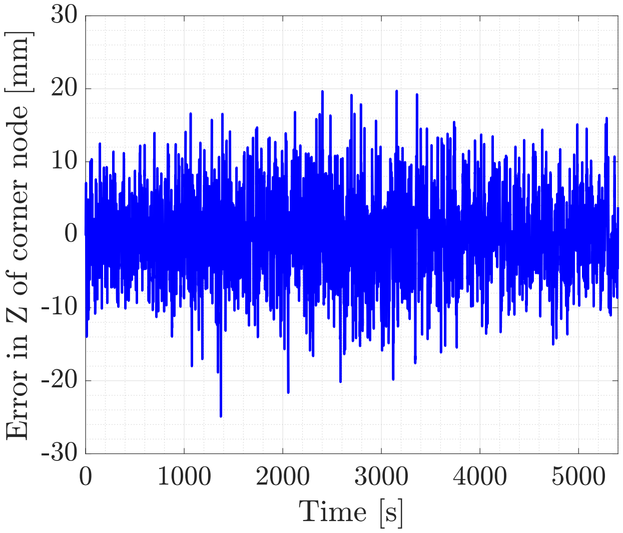


Figure 5.18: Evolution of difference between the $Z$-coordinate (inertial reference frame) of the estimated shape of the spacecraft calculated by the Kalman Filter using a $7 \times 7$ sensor array and the true shape of the spacecraft.

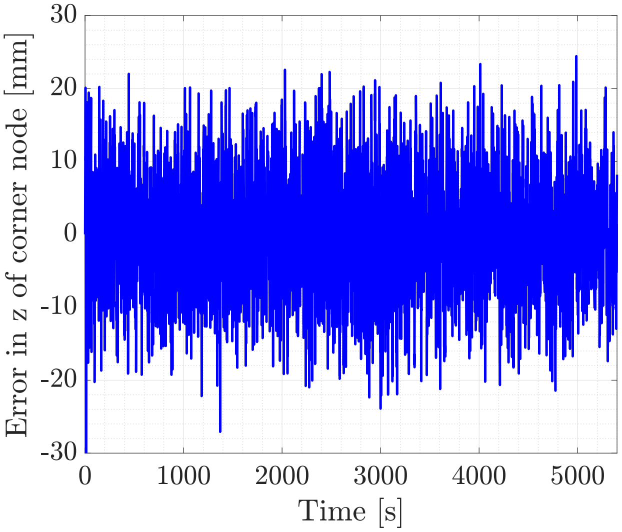


Figure 5.19: Evolution of difference between the $z$-coordinate (local reference frame) of the estimated shape of the spacecraft calculated by the Kalman Filter using a $7 \times 7$ sensor array and the true shape of the spacecraft.

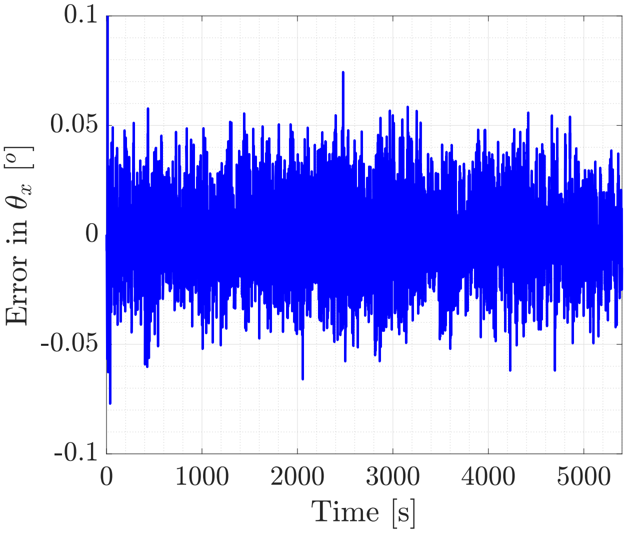


Figure 5.20: Evolution of difference between the rigid body angle of the estimated shape of the spacecraft calculated by the Kalman Filter using a $7 \times 7$ sensor array and the true shape of the spacecraft.

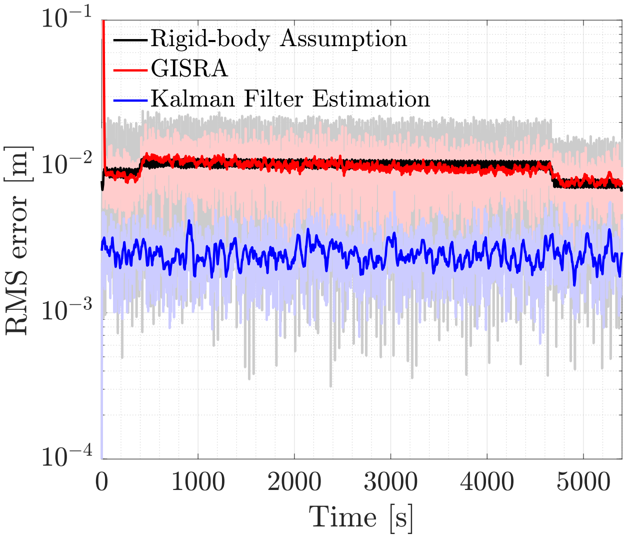


Figure 5.21: Evolution of the RMS error of the estimated shape of the spacecraft calculated by the Kalman Filter using a $7 \times 7$ sensor array (blue). The black curve shows the result of a model assuming the spacecraft as a rigid body and only measuring the rigid-body angle. The red curve is the result using the algorithm introduced in chapter 2. Dark colors represent 30s averages of the actual results in lighter color.

node of the model between the estimated shape and the true shape computed in a previous sub-section. The RMS of these errors is calculated for each time step and shown in figure 5.21. While it varies in function of time, it is relatively constant on average and is not affected by the force impulses. It is compared to a simple rigid model where the vibrations of the structure are discarded and only the rigid-body angle is measured with great accuracy, and the result of the shape reconstruction algorithm introduced in chapter 2 (named GISRA for Geometric Inextensible Shape Reconstruction Algorithm). Digitally filtered measurements were used for the later (using a 5th order yulewalk IIR filter with cutoff frequency of 0.2 Hz, above the main modes of excitations at 0.057 Hz and 0.091 Hz). The Kalman filter improves the accuracy of the shape estimation by an order of magnitude, hence proving its importance. The average RMS error of the reconstructed shape is about 2.5 mm. The performance of the initial algorithm is mainly limited by the phase shift due to the filter even if the noise of the measurmeent is reduced. Figure 5.22 shows an histogram of the RMS error as well as a fit by a Generalized Extreme Value distribution. This distribution fits the results very well and can be used in stochastic calculations of the performance of the spacecraft.

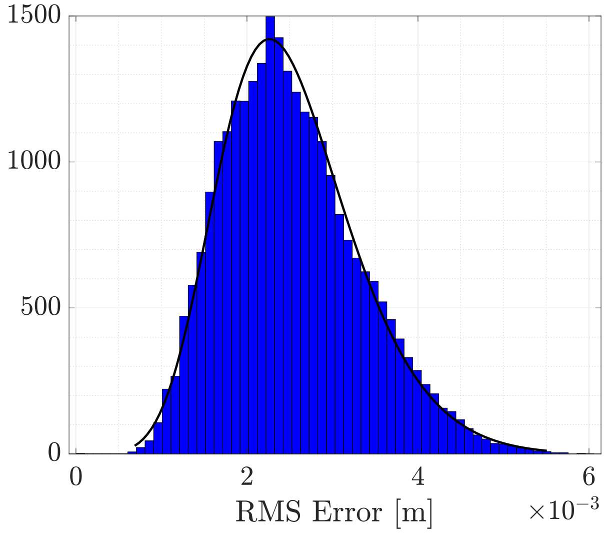


Figure 5.22: Histogram of the RMS error of the estimated shape of the spacecraft calculated by the Kalman Filter using a $7 \times 7$ sensor array. A fit by a Generalized Extreme Value Distribution is also plotted.

### Influence of the Number of Sensors

The number of sensors influences the accuracy of the Kalman filter estimation. An averaging effect appears where more measurements of a fixed number of variables reduce the noise of the calculated variables.

Figure 5.23 shows the average RMS error of the estimated shape for different numbers of sensors. The average is calculated from the RMS plot (see figure 5.21). The first 100 seconds of the simulations are discarded to make sure the filter has converged. A total of 10 simulations were performed for each data point to evaluate the $3\sigma$ confidence interval. Each data point corresponds to the following configurations of sensors: $2 \times 2$, $3 \times 3$, $4 \times 4$, $5 \times 5$, $7 \times 7$, $13 \times 13$, and $25 \times 25$.

As expected, the error decreases as the number of sensors increases which shows convergence of this solution. The number of sensors required to achieve millimeter level accuracy is reasonable, as just a few sensors are sufficient to get an accuracy of 5 mm. In this range, the curve can be approximated by a power function:

$$\mathrm{RMS}_{\mathrm{avg}} = \frac{10}{N_S^{0.35}} \quad [\mathrm{mm}] \tag{5.68}$$

where $N_S$ is the total number of sensors.

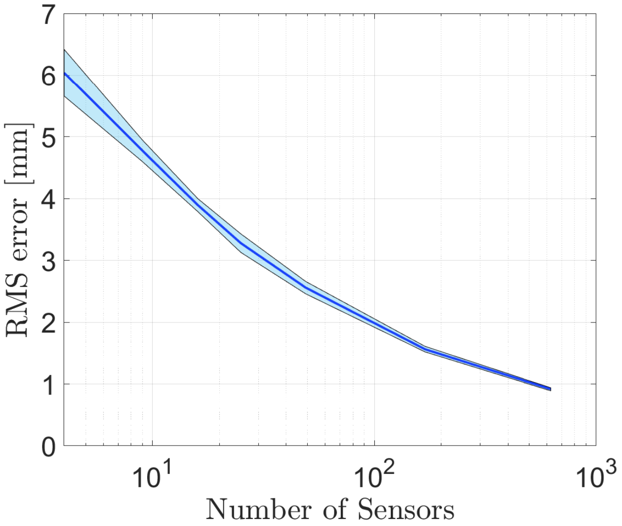


Figure 5.23: Evolution of the average RMS error in function of the number of sensors. The light blue region defines the $3\sigma$ confidence interval.

### Influence of the Error in the Estimation of the Stiffness

The true (simulated) shape of the structure and the estimated shape from the Kalman filter are both computed using the same finite element model. Even if the model for the Kalman filter is reduced, the stiffness and mass properties are identical.

In a real world application, there may be discrepancies between the actual stiffness of the structure and the one used in the computational models. While measuring the mass is rather simple, estimating the stiffness is more complicated, especially for complex structures.

Simulations are performed to show the robustness of the presented method under different stiffness errors. The finite element model used in the Kalman filter remain identical with homogeneous and isotropic properties and a bending stiffness of 14 Nm$^2$. The finite element model used to calculate the true shape of the spacecraft will have different stiffness properties.

First, we investigate the effect of varying the bending stiffness, while still using homogeneous and isotropic properties. The stiffness is varied from 7 Nm$^2$ to 28 Nm$^2$ (factors of 0.5 and 2 from the finite element model used in the Kalman filter 5.4). An array of $7 \times 7$ sensors is used for all simulations. Figure 5.24 shows the evolution of the average RMS error of the estimated shape in function of the stiffness ratio between the finite element model used to calculate the true response of the spacecraft and the one used in the Kalman filter. We see that it is minimum when the ratio is 1 and larger otherwise. The error is mostly dominated by the accumulated error between measurements resulting from the difference of frequency of the vibration modes.

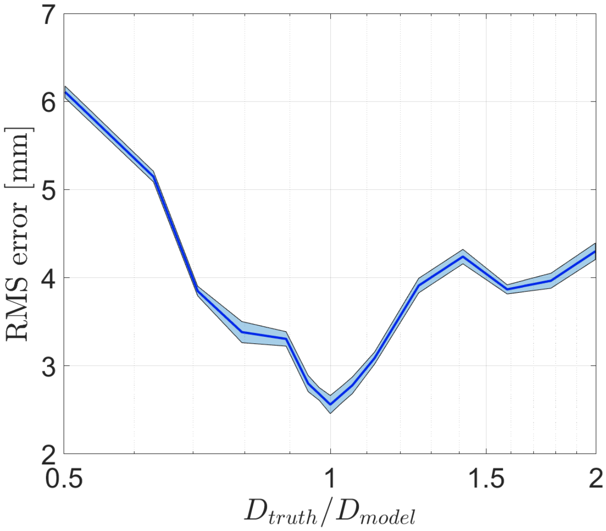


Figure 5.24: Evolution of the average RMS error in function of the ratio of bending stiffness used to calculate the true response of the spacecraft and the one used in the Kalman filter. A $7 \times 7$ array of sensors is used in the filter. The light blue region defines the $3\sigma$ confidence interval.

Second, we investigate the influence of the anisotropy of the stiffness matrix. The values of the bending stiffness in the $x$ and $y$ directions remain the same on average, but their ratio differs from 1. The stiffness matrix of the finite element model used to calculate the true response of the spacecraft becomes:

$$\mathbf{D} = D \begin{bmatrix} \frac{2\alpha}{1+\alpha} & 0 & 0 \\ 0 & \frac{2}{1+\alpha} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.69}$$

where $\alpha = \mathbf{D}_{11}/\mathbf{D}_{22}$. It is varied from 0.5 to 2. An array of $7{\times}7$ sensors is again used for all simulations. Figure 5.25 shows the evolution of the average RMS error of the estimated shape in function of the ratio $\alpha = \mathbf{D}_{11}/\mathbf{D}_{22}$. Similarly to the previous result, the RMS error gets worse as $\alpha$ differs from 1.

For both studies, the error increases and yet remains in the same order of magnitude even for relatively large discrepancies in the estimation of the stiffness parameters (factor of 2 between the actual and estimated values). This study can help the design process of such structures and inform the design team how accurate their model needs to be.

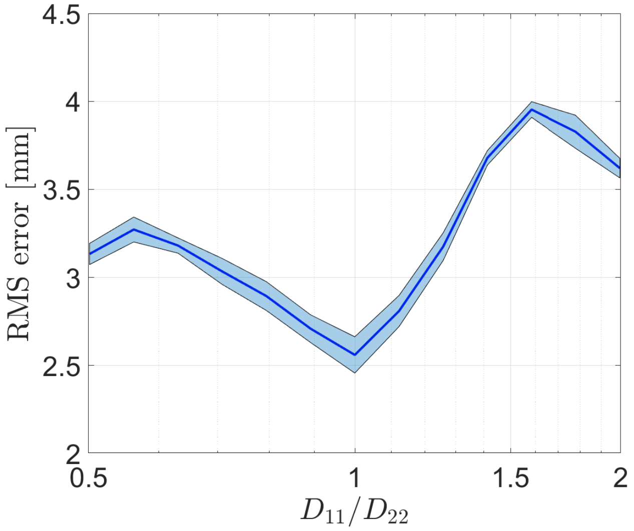


Figure 5.25: Evolution of the average RMS error in function of the ratio of bending stiffness in the $x$ and $y$ directions in the finite element model used to calculate the true response of the spacecraft. A 7 × 7 array of sensors is used in the Kalman filter. The light blue region defines the $3\sigma$ confidence interval.

### Sensor Fusion: Adding Force Sensors to the Sun Sensor System

To complete this study, we investigate the impact of adding different sensors to the Kalman filter. The simplicity of the formulation of the filter allows to add different kinds of sensors without changing the equations. The only requirement is for the measurement to be able to be derived from the state variables.

For instance, strain gauges can be added to measure local extensions or curvatures, RF antennas can be used to measure relative distance, etc.

We study the impact of adding force sensors to the current system. They measure the applied forces from the thrusters at each corner of the structure. Different systems can be used to measure millinewtons [36, 42, 63]. A 10% accuracy is selected for the sensors included in the simulation ($3\sigma$ error of 0.3 mN) which is representative of the state-of-the-art.

Figure 5.26 shows the RMS error of the estimated shape of the structure calculated by the Kalman filter integrating sun and force sensors. An array of 7 × 7 sun sensors is used and 4 force sensors (1 at each corner of the structure) measure the thruster impulses. The average RMS error is 1.3 mm, twice better than the solution without force sensors (figure 5.21) with only a little number of sensors added. More than 100 sun sensors would be needed to produce the same results (figure 5.23).

Note that in order for the Kalman filter to make the most of the force measurement, the process noise in the time derivative of the force is increased to $\mathbf{Q}_{\dot{F}} = 0.003^2 \times \mathbf{I}_{4\times4}$ [N$^2$/s$^2$].

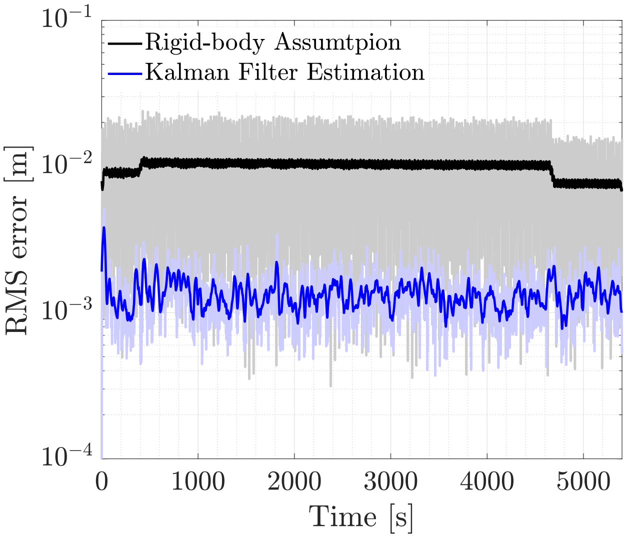


Figure 5.26: Evolution of the RMS error of the estimated shape of the spacecraft calculated by the Kalman Filter using a 7 × 7 sensor array and 4 force sensors (blue). The black curve shows the result of a model assuming the spacecraft as a rigid body and only measuring the rigid-body angle. Dark colors represent 30s averages of the actual results in lighter color.

## 5.6 Conclusion and Discussion

This chapter has demonstrated the feasibility and first order accuracy of a solution to measure the shape of a large space structure. Sun sensors are attached and distributed over the structure and measure the two local angles from the Sun. A Kalman filter that uses a dynamic mechanical model of the structure has been used to estimate the vibration and rigid-body angle of the structure.

This concept has been demonstrated by simulating the shape of a 25 × 25 m structure, which is a conceptual design for the Space Solar Power Project satellite under development at Caltech. The square spacecraft is maneuvered with 4 thrusters, 1 at each corner, and follows an optimal guidance path to maximize its functionality. Each time the thrusters fire, the satellite rotates and vibrations of few centimeters appear. The vibration amplitudes are accurately estimated by the presented algorithm/filter, and it is shown that millimeter level accuracy is achievable for such large structures.

Studies on the number of sensors, modelization errors on the stiffness of the structure and fusion with other type of sensors were presented. This gives first order results that provides an understanding of the behavior of the presented solutions and can be used to design future space missions.

A more detailed algorithm needs to be studied and validated experimentally if this is to be implemented on real hardware. The mechanical model presented in this chapter is simplified to make it linear. The main features were kept, that is the large rigid-body rotation and vibrations of the structure, but actual applications will require a more detailed model. This will likely result in non-linear equations both for the mechanical equilibrium and measurement. A simple Kalman filter such as the one presented in this chapter is no longer valid, and extended Kalman filter formulations will need to be considered. This leads to potential stability issues that will need to be addressed.

Furthermore, practical considerations such as the effects of time delays, especially when the number of sensors increases, have not be considered. Data communication and computation processing hardware may drive some limitations on the use and accuracy of the solution.

*Chapter 6*

# CONCLUSION

## 6.1 Summary and Contributions

The work presented in this thesis establishes the building blocks for a new type of shape measurement applicable to large space structures. By embedding distributed angle measurements directly on the surface to reconstruct, an alternative solution to optical based methods is introduced and is more suited for this type of application.

Current systems rely on external, optical solutions which usually limit the size and accuracy of the reconstructed shape. Some applications such as large planar spacecraft cannot rely on such systems. Measuring the shape of such satellites is vital for the expansion of the field of large space structures. Large solar arrays, solar sails and especially antenna arrays can benefit from the knowledge of the shape of the structure. Using this technology, new, ground breaking concepts can be imagined to push the limits of space applications. For instance, space solar power satellites, which have long been deemed impossible because of their mass and cost, can now be re-engineered using ultra-light support structures such as the Caltech Space Solar Power Project.

Space is not the only field where the presented solution can help with future applications. Wearable technologies, for instance, which can be used for medical, entertainment or multimedia purposes, can benefit from this system. Not relying on optical solutions to measure the shape of a person means that the system is not bounded to a specific location where cameras would be placed, for example.

In order to be able to show the functionality and performance of a system of embedded angle sensors, different steps have been undertaken and shown in this thesis. The three main contributions that have been made are as follows:

1. **The development of algorithms to integrate the distributed angle measurements into the shape of the support structure.** Different algorithms are introduced that can fit different applications. A geometry-based algorithm can reconstruct the static shape of most structures, but is limited by the smoothness of the estimated shape. A mechanically based algorithm has been proven more suited for structures that are subjected to more local deformations. Fi-

nally, a dynamic algorithm can estimate the vibrating shape of the structure using a Kalman filter. The filter combines the measurement of the sensors with a dynamic model of the structure to accurately estimate its shape.

2. **The experimental validation of the proposed solution.** A new type of sun sensor specifically suited for ultralight deployable space structures has been designed. It consists of a quad-photodiode underneath a square aperture. The full electronic and software design was performed in order to understand and solve the challenges associated with this new design. The calibration process was also detailed to understand the limitations of the current sensor and highlight the main areas of research required to continue its future development. The design of the experiment also highlighted the challenges and considerations that needs to be taken into account when implementing such a system. A 1.3 m long structure with 14 embedded sensors was used. Results of the experiment showed millimeter accuracy in the reconstruction of the shape of the structure and as low as 0.3 mm RMS.

3. **The beginning of the design process to apply the proposed solution to a planned space mission: the Caltech Space Solar Power Project Satellite.** A simplified mechanical model merging local vibrations of the structure and large rigid angles is presented and provides a basis for future models with similar applications. Initial studies have been conducted to get first order estimates on the accuracy of the solution. The density of sensors, errors in the mechanical models, and fusion of different sensors have been investigated. Such studies have demonstrated that, for example, an accuracy of 1 mm is achievable for a $25 \times 25$ m$^2$ structure with a $7 \times 7$ grid of sun sensors and 4 force sensors at the corners.

It is important to highlight the current limits of the proposed solution that have been identified through the different chapters of this thesis:

1. By embedding sensors directly on the surface to reconstruct, they become part of the design of the structure. This adds complexity to the overall system that is not present when using an external optical measurement tool. This solution is only advantageous when the measurement is vital to the application and when optical systems are not suited.

2. The reconstruction algorithms can only recreate relatively smooth surfaces. The system cannot measure highly complex shapes such as wrinkled surfaces. The periodicity of the deformation of the surface is limited by the spatial frequency of the sensors.

3. The more sensors, the more data needs to be communicated and analyzed. This can cause implementation issues as the hardware needs to be designed to enable communication over large distances and process large amount of measurements.

## 6.2 Future Work

Some areas of research are proposed to expand on this initial work. They are closely related to the current limits of the system and try to iteratively bring it closer to a real *commercial* application.

1. Decreasing the noise and error of the sensors has a direct impact on the accuracy of the solution. More work can be conducted to create more precise sun sensors while maintaining a relatively small package and mass. Different technologies or designs can be studied from the one presented in the thesis.

2. Relatively small deformations and smooth surfaces were studied in this thesis. Large deformations which could possibly shade few sensors might be of interest for some applications or simply to study the robustness of the solution. More complex geometries can be analyzed, especially structures with low bending stiffness that can create highly localized deformations.

3. Experimental demonstrations were limited to 14 sensors. For larger structures with a greater amount of sensors, implementations issues will arise. Specifically, a robust and fast communication architecture needs to be implemented without requiring heavy hardware.

4. A real-time version of the algorithms needs to be studied and experimented on spacecraft avionics. Low computational capabilities are available on a spacecraft and the actual implementation of the presented algorithms might be difficult.

# BIBLIOGRAPHY

[1] E. Abbena, S. Salamon, and A. Gray. *Modern Differential Geometry of Curves and Surfaces with Mathematica.* Chapman and Hall/CRC, 2017.

[2] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Vol. 55. US Government printing office, 1948.

[3] M. Arya, N. Lee, and S. Pellegrino. “Ultralight Structures for Space Solar Power Satellites.” In: *3rd AIAA Spacecraft Structures Conference*. 2016, p. 1950.

[4] J. Banik et al. “On-Orbit Validation of the Roll-Out Solar Array.” In: *2018 IEEE Aerospace Conference*. IEEE. 2018, pp. 1–9.

[5] J. Barnes, C. Liu, and K. Ariyur. “A Hemispherical Sun Sensor for Orientation and Geolocation.” In: *IEEE Sensors Journal* 14.12 (2014), pp. 4423–4433.

[6] J.-L. Batoz, K.-J. Bathe, and L.-W. Ho. “A Study of Three-Node Triangular Plate Bending Elements.” In: *International Journal for Numerical Methods in Engineering* 15.12 (1980), pp. 1771–1812.

[7] T. Belytschko et al. *Nonlinear Finite Elements for Continua and Structures.* John Wiley & Sons, 2013.

[8] M. Berger et al. “A Survey of Surface Reconstruction from Point Clouds.” In: *Computer Graphics Forum*. Vol. 36. 1. Wiley Online Library. 2017, pp. 301–329.

[9] C. P. Blankenship and R. J. Hayduk. *Large Space Structures-Structural Concepts and Materials.* Tech. rep. 872429. SAE Technical Paper, 1987.

[10] R. C. Boden and J. Hernando-Ayuso. “Shape Estimation of Gossamer Structures Using Distributed Sun-Angle Measurements.” In: *Journal of Spacecraft and Rockets* 55.2 (2017), pp. 415–426.

[11] F. Boldrini, E. Monnini, and Dorico Procopio. “Applications of APS Detector to GNC Sensors.” In: *4th IAA Symp. Small Satellites for Earth Observation*. 2003, pp. 33–40.

[12] F. Bosi et al. “Cure-Induced Deformation of Ultra-Thin Composite Laminates.” In: *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 2018, p. 2241.

[13] J. Brandrup et al. *Polymer handbook.* Vol. 89. John Wiley & Sons, 1999.

[14] R. G. Brown and P. Y. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* Vol. 3. Wiley New York, 1992.

[15] F. Brunet et al. “Monocular Template-Based Reconstruction of Smooth and Inextensible Surfaces.” In: *Asian Conference on Computer Vision*. Springer. 2010, pp. 52–66.

[16] J. H. Bruning et al. “Digital Wavefront Measuring Interferometer for Testing Optical Surfaces and Lenses.” In: *Applied optics* 13.11 (1974), pp. 2693–2703.

[17] A. W. Burner, W. A. Lokos, and D. A. Barrows. “In-flight Aeroelastic Measurement Technique Development.” In: *Optical Diagnostics for Fluids, Solids, and Combustion II*. Vol. 5191. International Society for Optics and Photonics. 2003, pp. 186–199.

[18] F. Chen, G. M. Brown, and M. Song. “Overview of 3-D Shape Measurement Using Optical Methods.” In: *Optical engineering* 39 (2000).

[19] F. Chen and J. Feng. “Analogue Sun Sensor Based on the Optical Nonlinear Compensation Measuring Principle.” In: *Measurement Science and Technology* 18.7 (2007), p. 2111.

[20] H. Contreras. “The Stochastic Finite-Element Method.” In: *Computers & Structures* 12.3 (1980), pp. 341–348.

[21] R. D. Cook. *Concepts and Applications of Finite Element Analysis.* John Wiley & Sons, 2007.

[22] M. Delapierre. “Dynamics and Stability of Spinning Membranes.” PhD thesis. California Institute of Technology, 2017.

[23] F. J. Delgado et al. “Accurate and Wide-Field-of-View MEMS-Based Sun Sensor for Industrial Applications.” In: *IEEE Transactions on Industrial Electronics* 59.12 (2012), pp. 4871–4880.

[24] P. Dierckx. *Curve and Surface Fitting with Splines.* Oxford University Press, 1995.

[25] Ł. Farian, P. Häfliger, and J. A. Leñero-Bardallo. “A Miniaturized Two-Axis Ultra Low Latency and Low-Power Sun Sensor for Attitude Determination of Micro Space Probes.” In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 65.5 (2017), pp. 1543–1554.

[26] C. A. Floudas and P. M. Pardalos. *Encyclopedia of Optimization.* Vol. 1. Springer Science & Business Media, 2001.

[27] B. Fornberg and J. Zuev. “The Runge Phenomenon and Spatially Variable Shape Parameters in RBF Interpolation.” In: *Computers & Mathematics with Applications* 54.3 (2007), pp. 379–398.

[28] J. Fraden. *Handbook of Modern Sensors: Physics, Designs, and Applications.* Springer Science & Business Media, 2004.

[29] R. Funase. “Attitude Control of IKAROS Solar Sail Spacecraft and its Flight Results.” In: *61st Int. Astronautical Congress*. 2010.

[30] E. E. Gdoutos et al. “Ultralight Spacecraft Structure Prototype.” In: *AIAA Scitech 2019 Forum*. 2019, p. 1749.

[31] E.E. Gdoutos et al. “A Lightweight Tile Structure Integrating Photovoltaic Conversion and RF Power Transfer for Space Solar Power Applications.” In: *2018 AIAA Spacecraft Structures Conference*. 2018, p. 2202.

[32] M. Géradin and D. J. Rixen. *Mechanical Vibrations: Theory and Application to Structural Dynamics.* John Wiley & Sons, 2014.

[33] R. Goldenthal et al. “Efficient Simulation of Inextensible Cloth.” In: *ACM Trans. Graph.* 26.3 (2007), p. 49.

[34] A. Goshtasby. “Design and Recovery of 2-D and 3-D Shapes Using Rational Gaussian Curves and Surfaces.” In: *International Journal of Computer Vision* 10.3 (1993), pp. 233–256.

[35] M. R. M. Hashemi et al. “A Flexible Phased Array System With Low Areal Mass Density.” In: *Nature Electronics* 2.5 (2019), p. 195.

[36] X. He et al. “A Submillimetric 3-DOF Force Sensing Instrument With Integrated Fiber Bragg Grating for Retinal Microsurgery.” In: *IEEE Transactions on Biomedical Engineering* 61.2 (2013), pp. 522–534.

[37] J. Hedgepeth. “Critical Requirements for the Design of Large Space Structures.” In: *2nd Conference on Large Space Platforms: Toward Permanent Manned Occupancy in Space*. 1981, p. 443.

[38] A. Hermanis, R. Cacurs, and M. Greitans. “Acceleration and Magnetic Sensor Network for Shape Sensing.” In: *IEEE Sensors Journal* 16.5 (2015), pp. 1271–1280.

[39] T. Hoshi and H. Shinoda. “3D Shape Measuring Sheet Utilizing Gravitational and Geomagnetic Fields.” In: *2008 SICE Annual Conference*. IEEE. 2008, pp. 915–920.

[40] M. Huard et al. “Reconstruction of Quasi Developable Surfaces From Ribbon Curves.” In: *Numerical Algorithms* 63.3 (2013), pp. 483–506.

[41] A. Ibrahimbegovic. “On the Choice of Finite Rotation Parameters.” In: *Computer Methods in Applied Mechanics and Engineering* 149.1-4 (1997), pp. 49–71.

[42] I. Iordachita et al. “A Sub-Millimetric, 0.25 mN Resolution Fully Integrated Fiber-Optic Force-Sensing Tool for Retinal Microsurgery.” In: *International journal of computer assisted radiology and surgery* 4.4 (2009), pp. 383–390.

[43] C. H. Jenkins. *Gossamer Spacecraft: Membrane and Inflatable Structures Technology for Space Applications.* American Institute of Aeronautics and Astronautics, 2001.

[44] A. Kay. *Operational Amplifier Noise: Techniques and Tips for Analyzing and Reducing Noise.* Elsevier, 2012.

[45] M. D. Kelzenberg et al. “Design and Prototyping Efforts for the Space Solar Power Initiative.” In: *2017 IEEE 44th Photovoltaic Specialist Conference*. IEEE. 2017, pp. 558–561.

[46] A. Khatamian and H. R. Arabnia. “Survey on 3D Surface Reconstruction.” In: *Journal of Information Processing Systems* 12.3 (2016).

[47] E. Kreyszig. *Introduction to Differential Geometry and Riemannian Geometry.* University of Toronto Press, 1968.

[48] H. Leclerc et al. “Integrated Digital Image Correlation for the Identification of Mechanical Properties.” In: *International Conference on Computer Vision/-Computer Graphics Collaboration Techniques and Applications*. Springer. 2009, pp. 161–171.

[49] F. Leens. “An Introduction to I$^2$C and SPI Protocols.” In: *IEEE Instrumentation & Measurement Magazine* 12.1 (2009), pp. 8–13.

[50] J. Leijtens, K. de Boom, and N. van der Heiden. “Micro Systems Technology: The Way to Shrink Sun Sensors.” In: *2005 International Conference on MEMS, NANO and Smart Systems*. IEEE. 2005, pp. 193–194.

[51] A. Lizotte and W. Lokos. “Deflection-Based Aircraft Structural Loads Estimation with Comparison to Flight.” In: *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. 2005, p. 2016.

[52] L. E. Malvern. *Introduction to the Mechanics of a Continuous Medium.* Monograph. 1969.

[53] M. A. Marshall, A. Goel, and S. Pellegrino. “Attitude Maneuver Design for Planar Space Solar Power Satellites.” 2019.

[54] D. Metaxas and D. Terzopoulos. “Constrained Deformable Superquadrics and Nonrigid Motion Tracking.” In: *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 1991, pp. 337–343.

[55] M. O’hara. *EMC at Component and PCB Level.* Elsevier, 1998.

[56] J.-J. Orteu et al. “A Speckle Texture Image Generator.” In: *Speckle06: Speckles, From Grains to Flowers*. Vol. 6341. International Society for Optics and Photonics. 2006, 63410H.

[57] A. Pelemeshko et al. “High-Precision CubeSat Sun Sensor Coupled with Infrared Earth Horizon Detector.” In: *IOP Conference Series: Materials Science and Engineering*. Vol. 734. 1. IOP Publishing. 2020, p. 012012.

[58] K. Perlin. “An Image Synthesizer.” In: *ACM Siggraph Computer Graphics* 19.3 (1985), pp. 287–296.

[59] M. Perriollat, R. Hartley, and A. Bartoli. “Monocular Template-Based Reconstruction of Inextensible Surfaces.” In: *International Journal of Computer Vision* 95.2 (2011), pp. 124–137.

[60] Zach Peterson. “Second Generation Wallops Fine Sun Sensor.” In: (2016).

[61] M. A. Post, J. Li, and R. Lee. “A Low-Cost Photodiode Sun Sensor for CubeSat and Planetary Microrover.” In: *International Journal of Aerospace Engineering* 2013 (2013).

[62] R. S. Quimby. “Photonics and Lasers.” In: *John Wiley & Sons* (2006).

[63] B. W. Quist and M. J. Hartmann. “A Two-Dimensional Force Sensor in the Millinewton Range for Measuring Vibrissal Contacts.” In: *Journal of neuroscience methods* 172.2 (2008), pp. 158–167.

[64] P. K. Rastogi. *Photomechanics.* Vol. 77. Springer Science & Business Media, 2003.

[65] P. Regalia. *Adaptive IIR Filtering in Signal Processing and Control.* Routledge, 2018.

[66] M.M.G. Ricci and T. Levi-Civita. “Méthodes de calcul différentiel absolu et leurs applications.” In: *Mathematische Annalen* 54.1-2 (1900), pp. 125–201.

[67] L. Rios-Reyes and D. J. Scheeres. “Solar-Sail Navigation: Estimation of Force, Moments, and Optical Parameters.” In: *Journal of Guidance, Control, and Dynamics* 30.3 (2007), pp. 660–668.

[68] D. J. Rixen. “A Dual Craig–Bampton Method for Dynamic Substructuring.” In: *Journal of Computational and applied mathematics* 168.1-2 (2004), pp. 383–391.

[69] F. Royer and S. Pellegrino. “Buckling and Post-Buckling of Ladder-Type Coilable Space Structures.” 2019.

[70] E. J. Ruggiero and D. J. Inman. “Gossamer Spacecraft: Recent Trends in Design, Analysis, Experimentation, and Control.” In: *Journal of Spacecraft and Rockets* 43.1 (2006), pp. 10–24.

[71] L. Salgado-Conrado. “A Review on Sun Position Sensors Used in Solar Applications.” In: *Renewable and Sustainable Energy Reviews* 82 (2018), pp. 2128–2146.

[72] M. Salzmann et al. “Closed-Form Solution to Non-Rigid 3D Surface Registration.” In: *European Conference on Computer Vision*. Springer. 2008, pp. 581–594.

[73] M. Salzmann et al. “Surface Deformation Models for Nonrigid 3D Shape Recovery.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.8 (2007), pp. 1481–1487.

[74] H. Sawada et al. “Shape Estimation of IKAROS’s Solar Power Sail by Images of Monitor Cameras.” In: *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*. 2012, p. 1749.

[75] R. W. Schock. “Solar Array Flight Dynamic Experiment.” In: *Workshop on Structural Dynamics and Control Interaction of Flexible Structures*. 1986, pp. 487–504.

[76] A. A. Shabana. *Dynamics of Multibody Systems.* Cambridge university press, 2013.

[77] A. A. Shabana and A. P. Christensen. “Three-dimensional Absolute Nodal Coordinate Formulation: Plate Problem.” In: *International journal for numerical methods in engineering* 40.15 (1997), pp. 2775–2790.

[78] J. Smisek, M. Jancosek, and T. Pajdla. “3D With Kinect.” In: *Consumer Depth Cameras for Computer Vision*. Springer, 2013, pp. 3–25.

[79] E. D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems.* Vol. 6. Springer Science & Business Media, 2013.

[80] N. Sprynski, B. Lacolle, and L. Biard. “Motion Capture of an Animated Surface via Sensors’ Ribbons.” In: *PECCS 2011-1st International Conference on Pervasive and Embedded Computing and Communication Systems*. SciTePress. 2011, pp. 421–426.

[81] T. Stanko et al. “Shape From Sensors: Curve Networks on Surfaces From 3D Orientations.” In: *Computers & Graphics* 66 (2017), pp. 74–84.

[82] J. Steeves et al. “Multilayer Active Shell Mirrors for Space Telescopes.” In: *Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation II*. Vol. 9912. International Society for Optics and Photonics. 2016, 99121K.

[83] M. A. Sutton, J. J. Orteu, and H. Schreier. *Image Correlation for Shape, Motion and Deformation Measurements: Basic Concepts, Theory and Applications.* Springer Science & Business Media, 2009.

[84] T. Talon and S. Pellegrino. “In-Space Shape Measurement of Large Planar Structures.” In: *4th AIAA Spacecraft Structures Conference*. 2017, p. 1116.

[85] T. Talon and S. Pellegrino. “Shape Measurement of Large Structures in Space: Experiments.” In: *2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. IEEE. 2018, pp. 581–584.

[86] L. Torresani, A. Hertzmann, and C. Bregler. “Learning Non-Rigid 3D Shape From 2D motion.” In: *Advances in Neural Information Processing Systems*. 2004, pp. 1555–1562.

[87] A. Trebi-Ollennu et al. “Design and Analysis of a Sun Sensor for Planetary Rover Absolute Heading Detection.” In: *IEEE Transactions on Robotics and Automation* 17.6 (2001), pp. 939–947.

[88] Y. Tsuda et al. “Achievement of IKAROS—Japanese Deep Space Solar Sail Demonstration Mission.” In: *Acta Astronautica* 82.2 (2013), pp. 183–188.

[89] Y. Tsuda et al. “Shape Parameters Estimation of IKAROS Solar Sail Using In-Flight Attitude Determination Data.” In: *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 19th AIAA/ASME/AHS Adaptive Structures Conference 13th AIAA*. 2011, p. 1889.

[90] H. Tsunoda and Y. Senbokuya. “Shape Measurement of a Flat Stretched Lightweight Membrane for Planar Antenna Structure.” In: *19th AIAA Applied Aerodynamics Conference*. 2001, p. 1597.

[91] G. Wang et al. “Precision Enhancement Method for Multiplexing Image Detector-Based Sun Sensor With Varying and Coded Apertures.” In: *Applied optics* 54.35 (2015), pp. 10467–10472.

[92] Y. Wei. “Deployable Piezoelectric Thin Shell Structures: Concepts, Characterization and Vibration Control.” PhD thesis. California Institute of Technology, 2019.

[93] D. J. Whitehouse. *Surfaces and Their Measurement.* Elsevier, 2004.

[94] N. Xie and A. J.P. Theuwissen. “Low-Power High-Accuracy Micro-Digital Sun Sensor by Means of a CMOS Image Sensor.” In: *Journal of Electronic Imaging* 22.3 (2013), p. 033030.

[95] G. Xu et al. “Design of PSD Based Solar Direction Sensor.” In: *Sixth International Symposium on Precision Mechanical Measurements*. Vol. 8916. International Society for Optics and Photonics. 2013, 89162K.

[96] Giora Yahav, Gabi J Iddan, and David Mandelboum. “3D Imaging Camera for Gaming Application.” In: *2007 Digest of Technical Papers International Conference on Consumer Electronics*. IEEE. 2007, pp. 1–2.

[97] J. Yang et al. “Vision-based Localization and Robot-centric Mapping in Riverine Environments.” In: *Journal of Field Robotics* 34.3 (2017), pp. 429–450.

[98] S. Zhang and P. S. Huang. “High-Resolution, Real-Time Three-Dimensional Shape Measurement.” In: *Optical Engineering* 45.12 (2006), p. 123601.

[99] Z. Zhang. “Microsoft Kinect Sensor and its Effect.” In: *IEEE Multimedia* 19.2 (2012), pp. 4–10.